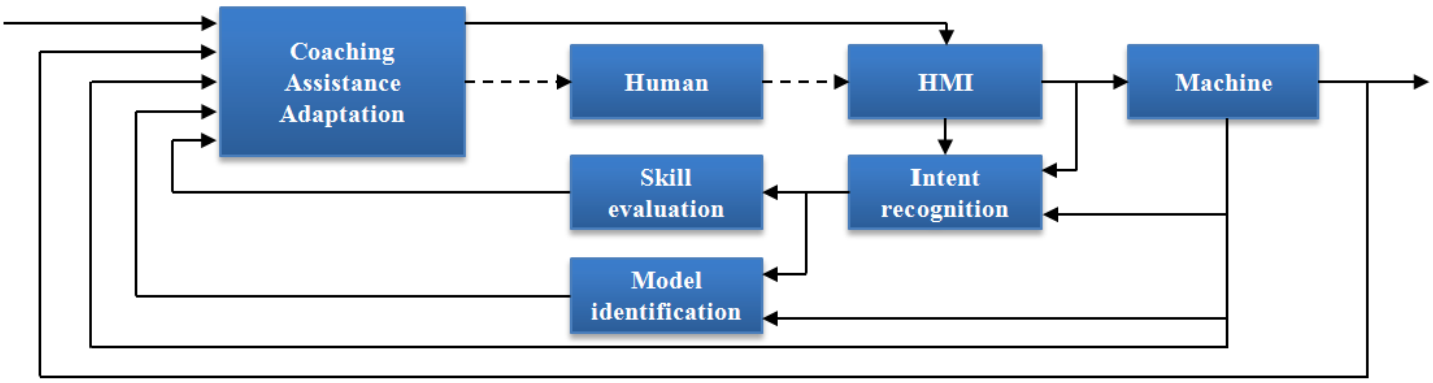
Helsinki University of Technology Control Engineering

Espoo 2010 Report 168

HUMAN ADAPTIVE MECHATRONICS METHODS FOR MOBILE WORKING MACHINES

Kalevi Tervo



Aalto University
School of Science
and Technology

Helsinki University of Technology Control Engineering

Espoo November 2010 Report 168

HUMAN ADAPTIVE MECHATRONICS METHODS FOR MOBILE WORKING MACHINES

Kalevi Tervo

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Electronics, Communications and Automation for public examination and debate in Auditorium AS1 at the Aalto University School of Science and Technology (Espoo, Finland) on the 18th of December 2010 at 10 a.m.

Aalto University

School of Science and Technology

Faculty of Electronics, Communications and Automation

Department of Automation and Systems Technology

Distribution:
Aalto University
Department of Automation and Systems Technology
P.O. Box 15500
FI-00076 Aalto, Finland
Tel. +358-9-470 25201
Fax. +358-9-470 25208
E-mail: control.engineering@tkk.fi
http://autsys.tkk.fi/

ISBN 978-952-60-3529-1 (printed)
ISBN 978-952-60-3530-7 (pdf)
ISSN 0356-0872

URL: http://lib.tkk.fi/Diss/2010/isbn9789526035307

Aalto-Print
Helsinki 2010

Aalto University

ABSTRACT OF DOCTORAL DISSERTATION	AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY P.O. BOX 11000, FI-00076 AALTO http://www.aalto.fi

Author Kalevi Tervo

Name of the dissertation
Human adaptive mechatronics methods for mobile working machines

Manuscript submitted 2.7.2010	Manuscript revised 29.11.2010
Date of the defence 18.12.2010	
☒ Monograph	☐ Article dissertation (summary + original articles)

Faculty	Faculty of Electronics, Communications and Automation
Department	Department of Automation and Systems Technology
Field of research	Control Engineering
Opponent(s)	Prof. Juha T. Tanttu, Prof. Hannu Koivisto
Supervisor	Prof. Heikki Koivo
Instructor	Prof. Heikki Koivo

Abstract

Despite the trend of increasing automation degree in control systems, human operators are still needed in applications such as aviation and surgery, or machines used in remote mining, forestry, construction, and agriculture, just to name a few. Although there are research results showing that the performance between the operators of working machines differ significantly, there are currently no means to improve the performance of the human-machine system automatically based on the skill and working differences of the operators.

Traditionally the human-machine systems are designed so that the machine is “constant” for every operator. On the contrary, the Human Adaptive Mechatronics (HAM) approach focuses on individual design, taking into account the skill differences and preferences of the operators. This thesis proposes a new type of a HAM system for mobile working machines called Human Adaptive Mechatronics and Coaching (HAMC) system that is designed to account for the challenges regarding to the measurement capability and the work complexity in the real-life machines. Moreover, the related subproblems including intent recognition, skill evaluation, human operator modeling, intelligent coaching and skill adaptivity are described.

The intent recognition is solved using a Hidden Markov model (HMM) based work cycle modeling method, which is a basis for the skill evaluation. The methods are implemented in three industrial applications. The human operator modeling problem is studied from the structural models’ perspective. The structural models can be used to describe a continuum of human operator models with respect to the operating points of the controlled machine. Several extensions and new approaches which enable more efficient parameter estimation using the experimental data are described for the conventional Modified Optimal Control Model (MOCM) of human operator. The human operator modeling methods are implemented to model a human operator controlling a trolley crane simulator. Finally, the concept of human adaptive Human-Machine Interface (HMI) is described. The analytic and knowledge-based approaches for realizing the HMI adaptation are presented and implemented for trolley crane simulator control.

Keywords Human adaptive mechatronics, skill evaluation, human modeling, human-machine systems

ISBN (printed) 978-952-60-3529-1	ISSN (printed) 0356-0872
ISBN (pdf) 978-952-60-3530-7	ISSN (pdf)
Language English	Number of pages 230 p.

Publisher Aalto University, Department of Automation and Systems Technology

Print distribution Aalto University, Department of Automation and Systems Technology

☒ The dissertation can be read at http://lib.tkk.fi/Diss/2010/isbn9789526035307

Aalto-yliopisto

VÄITÖSKIRJAN TIIVISTELMÄ	AALTO-YLIOPISTO TEKNILLINEN KORKEAKOULU PL 11000, 00076 AALTO http://www.aalto.fi
Tekijä Kalevi Tervo	
Väitöskirjan nimi Ihmisadaptiivisen mekatroniikan menetelmiä liikkuville työkoneille	
Käsikirjoituksen päivämäärä 2.7.2010	Korjatun käsikirjoituksen päivämäärä 29.11.2010
Väitöstilaisuuden ajankohta 18.12.2010	
☒ Monografia	☐ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Tiedekunta	Elektroniikan, tietoliikenteen ja automaation tiedekunta
Laitos	Automaatio- ja systeemitekniikan laitos
Tutkimusala	Systeemitekniikka
Vastaväittäjä(t)	Prof. Juha T. Tanttu, Prof. Hannu Koivisto
Työn valvoja	Prof. Heikki Koivo
Työn ohjaaja	Prof. Heikki Koivo

Tiivistelmä

Viimeaikaisesta automaatioasteen lisääntymisestä huolimatta ihmisen rooli prosessien ja koneiden ohjauksessa on edelleen merkittävä. Ihmisoperaattoreita tarvitaan esimerkiksi lentokoneissa, robottiavusteisessa kirurgiassa, sekä kaivos-, metsä- ja maataloustyökoneissa. Tutkimusten mukaan operaattoreiden välillä on tyypillisesti merkittäviä suorituskykyeroja. Tästä huolimatta nykyisten koneiden tai kuljettajien toimintaa ei voida automaattisesti parantaa taito- ja toimintaerojen perusteella.

Perinteisesti ihminen-kone-järjestelmiä suunnitellaan siten, että ne toimivat mahdollisimman hyvin kaikilla käyttäjillä. Ihmisadaptiivisen mekatroniikan (HAM) näkökulma on päinvastainen. Siinä pyritään mukauttamaan kone tai prosessi siten, että se on mahdollisimman hyvä yksittäiselle käyttäjälle ottaen huomioon käyttäjän taitotason ja mieltymykset. Tässä väitöskirjassa esitetään uudenlainen HAM järjestelmä, jota kutsutaan ihmisen toimintaan mukautuvaksi valmennusjärjestelmäksi (HAMC). Järjestelmä on suunniteltu ottamaan huomioon erityisesti työkoneiden haasteelliset mittausolosuhteet ja monimutkaiset työtehtävät. Lisäksi esitellään seuraavat aihepiiriin liittyvät haasteet: ihmisen aikomuksen tunnistaminen, taidon arviointi, ihmissäätimen mallintaminen, älykäs valmennus ja ihminen-kone-rajapinnan mukauttaminen.

Tässä väitöskirjassa esitetään uusi menetelmä ihmisen tekemän työn tehtävien tunnistamiseen ja luokitteluun Markovin piilomallin (HMM) avulla, sekä esitellään siihen perustuva taidon arviointimenetelmä. Nämä menetelmät toteutetaan kolmessa teollisuussovelluksessa. Väitöskirjassa tutkitaan ihmisen mallinnusta rakenteellisten mallien näkökulmasta, koska ne mahdollistavat mallin mukauttamisen ohjattavan koneen toimintapisteiden funktiona. Perinteiseen optimisäätömalliin (MOCM) esitetään useita laajennuksia ja uusia näkökulmia, joiden avulla mallin parametreja voidaan estimoida paremmin kokeellisesta datasta. Ihmisen mallinnusmenetelmiä testataan yksinkertaisella nosturisimulaattorilla kerätyllä datalla. Viimeisessä luvussa esitetään adaptiivisen ihminen-kone-rajapinnan rakenne, työssä tutkitut viritys- ja adaptoitumismenetelmät sekä kokeelliset tulokset

Asiasanat	Ihmisadaptiivinen mekatroniikka, taidon arviointi, ihmisen mallintaminen, ihminen-kone-järjestelmät		
ISBN (painettu)	978-952-60-3529-1	ISSN (painettu)	0356-0872
ISBN (pdf)	978-952-60-3530-7	ISSN (pdf)	
Kieli	Englanti	Sivumäärä	230 s.
Julkaisija	Aalto-yliopisto, Automaatio- ja systeemitekniikan laitos		
Painetun väitöskirjan jakelu	Aalto-yliopisto, Automaatio- ja systeemitekniikan laitos		
☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2010/isbn9789526035307			

Preface

This thesis presents the research work I have done during 2007–2010 at the Control Engineering Research Group (former Control Engineering Laboratory) of the Department of Automation and Systems Technology, at Aalto University School of Science and Technology. To be exact, until 2010 the name of the university was Helsinki University of Technology (TKK).

My research work was supervised by Professor Heikki Koivo, to whom I owe my gratitude for giving me the opportunity to do research in his group and having the faith to let me work on the area nobody in our group had previously worked. Professor Koivo has an amazing ability to encourage his students and workers to achieve things which they might themselves feel impossible. That is academic leadership at its best.

The pre-examination of the thesis manuscript was done by Professor Sandra Hirche (Technische Universistät München) and Professor Satoshi Suzuki (Tokyo Denki University). I truly appreciate their effort and critical yet encouraging comments, which significantly improved the thesis.

The research work for the thesis has been carried out under several projects. It started in the LATU subproject of the research project AULA (2007–2008) funded by the Finnish Funding Agency for Technology and Innovation (TEKES) and a consortium of Finnish companies. In the beginning of 2008 a grant was given for our group by the Centennial Foundation of the Federation of Finnish Technology Industries for the research project OTTO (2008–2010). The thesis was finished in the project INTO (2009–2010) funded by TEKES and industry.

Research-wise, I owe my greatest gratitude for Mr. Lauri Palmroth (Tampere University of Technology), an ideamachine and a character who has set me an example and given inspiration during the work. Without you this research work would not have began in the first place. The beginning of my research career was supported and guided by Dr. Vesa Hölttä to whom I am grateful for teaching me a lot about research, scientific writing and project management. Between theory and practice there has been Dr. Jani Kaartinen who I acknowledge of his role in the implementation of the research results in forest machines.

In OTTO and INTO projects, I had a possibility to serve as a project manager and to lead my own talented group of researchers. Throughout these projects, Mr. Samuel Aulanko played a vital role in the work cycle modeling of industrial cranes. The practical implementation of the models would not have been possible without the effort of Mr. Anssi Pekkola and Mr. Joonas Varso. Joonas was also the key person in the work cycle modeling of overhead cranes. In the human operator modeling, I am grateful for the work of Ms. Aino Manninen who I supervised during several summers. Samuel, Anssi, Joonas and Aino: Thank you, the crane research would not have been so smooth and successful without your expertise and enthusiasm.

In autumn 2008 I got a unique opportunity to visit the Research Center for Advanced Technologies (RCAT) of Tokyo Denki University (TDU). The visit position was arranged by the TDU President, Professor Katsuhisa Furuta and the RCAT Director Professor Iku Nemoto. Thank you for making the visit possible. At RCAT I collaborated with Professor Atsushi Aoyama to whom I am grateful for giving me an introduction to the area of brain research, as well as the memorable conversations and dinners we had during the three months.

During the research work I received funding from several sources. In addition to the projects

funded by TEKES and industry, the graduate school of the TKK Faculty of Electronics, Communications and Automation provided continuous support for the research. Additional support has been given by Finnish Cultural Foundation Kainuu Regional fund, Finnish Foundation for Technology Promotion, Walter Ahlström Foundation, Wihuri fund, the Finnish Foundation for Economic and Technology Sciences – KAUTE, Henry Ford Foundation, and the Finnish Society for Automation. All this support is greatly acknowledged.

For the soul of an engineer it is the practice that gives the inspiration. Most of the research problems discussed in this thesis have emerged in practical applications. The long collaboration in development of forest machines with John Deere Forestry has enabled unique practical implementations of the research results in real-life operation. This would not have been possible without the enthusiasm of Dr. Arto Peltomaa and Mr. Aki Putkonen. Without your positive and open attitude towards new ideas, our research around intelligent human-machine systems would most likely not have begun in the first place. You truly understand what long term research is all about. The other application area of this thesis is industrial cranes. In that field the cooperation between Konecranes has been vital. I want to thank Dr. Timo Sorsa for his guidance and collaboration during OTTO and INTO projects.

I have worked at the Control Engineering group since the summer 2005. This time has been memorable. I have learned the importance of inspiring atmosphere and colleagues. Along with the ones I did not mention previously, I want to emphasize the role of Kai Zenger, Lasse Eriksson, Maurizio Bocca, Olli Haavisto, Ali Pekcan, Antti Pohjoranta, Juha Orivuori, Antti Remes and Janne Pietilä for setting examples and being continuous sources of inspiration. Just a quick conversation with you can solve a problem which I might have tried to solve for weeks. The whole staff of Control Engineering Group, thank you. Because of you all, I have been happy (almost) every morning I have arrived to work.

During the years, the support from my family and friends has been invaluable. I am grateful for my parents, brothers, sisters and friends for standing by me and giving me balance between the research life and the other things in life. Finally I want to thank my lovely (future) wife Suvi for her continuous and unconditional support and understanding during the process of writing this thesis. It is all because of you, (almost) every evening I have returned home, I have been happier than in the morning when I went to work.

Kalevi Tervo

Contents

List of Acronyms

ANFIS Adaptive Network-Based Neuro-Fuzzy Inference System

ARMA Autoregressive Moving Average

ARMAX Autoregressive Moving Average with Exogenous Input

ARX Autoregressive with Exogenous Input

CAN Controller Area Network

DBN Dynamic Bayesian Network

DBSFM Data-Based State Feedback Model

DDBSFM Discrete Data-Based State Feedback Model

FDI Fault Detection and Identification

FFT Fast Fourier Tranformation

FHMM Factorial Hidden Markov Model

FIS Fuzzy Inference System

FTC Fault Tolerant Control

FOMOCM Fixed-Order Modified Optimal Control Model

GEV Generalized Extreme Value Distribution

GEVP Generalized Eigenvalue Problem

GPDE Gradient Propagation Difference Equation

HAM Human Adaptive Mechatronics

HAMC Human Adaptive Mechatronics and Coaching

HDBN Hybrid Dynamic Bayesian Network

HHM Harvester Head Module

HHMM Hybrid Hidden Markov Model

HMCS Human-Machine Collaborative Systems

HMI Human-Machine Interface

HMM Hidden Markov Model

HWDDBSFM Hammerstein-Wiener DDBSFM

HWMOCM Hammerstein-Wiener MOCM

HWOE Hammerstein-Wiener Output Error Model

ICS Intelligent Coaching System

InSQL Industrial SQL

IQR Interquartile Range

ISE Integral of Squared Error

ITS Intelligent Tutoring System

JND Just Noticiable Difference

LHMM Layered Hidden Markov Model

LMI Linear Matrix Inequality

LMIP LMI problem

LogP Logarithmic probability of the observation sequence given the HMM

LQG Linear Quadratic Gaussian

LQR Linear Quadratic Regulator

MIMO Multi Input Multi Output

MISO Multi Input Single Output

MOCM Modified Optimal Control Model

MsSQL Microsoft SQL

NARMAX Nonlinear Autoregressive Moving Average with Exogenous Input

OCM Optimal Control Model

PLC Programmable Logic Controller

RDBMS Relational Database Management System

RMG Rail Mounted Gantry Crane

RT Road Truck

RTG Rubber Tired Gantry Crane

SAC Skill Adaptive Control

SC Straddle Carrier

SOM Self Organizing Maps

SQL Structured Query Language

STS Ship-to-Shore Container Crane

StateP Logarithmic state probability

XML Extensible Markup Language

List of Notations

Operators and functions

$\hat{x}$	Estimated value of x
x^*	Optimal value of x
x^H	Hermitian operation for complex number x (or matrix)
$\mathbf{x}^T$	Transpose of matrix or vector $\mathbf{x}$
$\dot{x}$	Derivative of x with respect to time
$\arg(x)$	Argument of complex number x
$\arg\min_a(x(a))$	The value of variable a resulting in minimum value of x
$\det(\mathbf{X})$	Determinant of matrix $\mathbf{X}$
$\mathrm{diag}(x_1, x_2, \ldots, x_n)$	Diagonal matrix $\begin{bmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & x_n \end{bmatrix}$
$e^x,\ \exp(x)$	Exponent function with x
$\mathrm{Re}(x)$	Real part of complex number x
SAT	Ideal symmetric saturation function
sgn	Signus function
$\mathrm{tr}(\mathbf{X})$	Trace of matrix $\mathbf{X}$
$\frac{\partial}{\partial\theta_l}$	Partial derivative with respect to θ_l
$\delta_{i,j}$	Two argument Kronecker delta

Roman symbols

a_{ij}	Probability to move from state S_i to state S_j in terms of HMMs.
a_n	Discrete time neuromotor time constant
$\mathbf{A}$	$\mathbf{A} = \{a_{ij}\}$ one step state transition probability distribution in terms of HMMs. System matrix for continuous state-space representation.
$\tilde{\mathbf{A}}$	Closed-loop human-machine system in terms of MOCM
$\mathbf{A},\ \mathbf{B},\ \mathbf{C},\ \mathbf{D},\ \mathbf{E}$	Matrices of a continuous time plant
$\mathbf{A}_0,\ \mathbf{B}_0,\ \mathbf{C}_0,\ \mathbf{E}_0$	Matrices of 0-plant
$\mathbf{A}_1,\ \mathbf{B}_1,\ \mathbf{C}_1,\ \mathbf{E}_1$	Matrices of 1-plant
$\mathbf{A}_{1,n}$	Modified 1-plant
$\mathbf{A}_d,\ \mathbf{B}_d,\ \mathbf{C}_d$	Matrices of state-space representation of the Padé approximation
$\mathbf{A}_\mathrm{g},\ \mathbf{B}_\mathrm{g}$	Initial guesses for HMM matrices
$\tilde{\mathbf{A}}_k$	Shorthand notation for $i\omega_k\mathbf{I} - \mathbf{A}_p$
$\mathbf{A}_p,\ \mathbf{B}_p,\ \mathbf{C}_p,\ \mathbf{E}_p$	Matrices of human operator model
$\mathbf{A}_s,\ \mathbf{B}_s,\ \mathbf{C}_s,\ \mathbf{D}_s$	Matrices of s-plant
$\mathbf{A}_\mathrm{sup},\ \mathbf{B}_\mathrm{sup}$	HMM matrices given by supervised training
$\mathbf{A}_\mathrm{uns},\ \mathbf{B}_\mathrm{uns}$	HMM matrices given by unsupervised training

$b,\ L,\ g,\ m,\ M$	Linear friction coefficient, rope length, gravitation coefficient, load mass, and trolley mass of the trolley crane simulator model
$b_i(k)$	Observation probability for character v_k in state S_i
$\mathbf{B}$	$\mathbf{B} = \{b_i(k)\}$ Observation probability distribution in terms of HMMs. Input matrix for continuous state-space representation.
$\tilde{\mathbf{B}}_k$	Shorthand notation for $\mathbf{B}_p\mathbf{Y}_k$
C_f	Return difference condition constraint for state estimator
C_r	Return difference condition constraint for state feedback controller
CM	Confusion matrix
$\mathbf{C}_a,\ \mathbf{D}_a$	Assumed matrices for human operator internal plant
$\tilde{\mathbf{C}}_k$	Shorthand notation for $e^{-\tau_p i\omega_k}\mathbf{C}_p$
$\mathbf{C}_{sa}$	Assumed matrices for human operator s-plant
$\mathbf{D}_1$	Temporary matrix used in LMI solution for MOCM control and estimator gains
E_i	Efficiency of task S_i
f	Control rate weighting in the optimal control cost function
f_i	Value of interest in task S_i
f_{tot}	Sum of the task-wise f_i values for a work cycle
$\mathbf{F}$	State estimator gain matrix
g_i	The i^{th} element of $\mathbf{g}_p$
$g_i(\mathbf{x}(t), \boldsymbol{\xi})$	Model of HMI
$g_m(\mathbf{x}(t))$	Model of controlled machine
$g_p(\mathbf{x}(t))$	Model of human operator
$G_f(s)$	Return difference transfer function for state estimator
$G_i(s, \boldsymbol{\xi})$	The transfer function of HMI
$G_m(s)$	Transfer function of controlled machine
$G_p(s)$	Human operator transfer function
$G_r(s)$	Return difference transfer function for state feedback controller
$\mathbf{g}_p$	State feedback gain matrix of 0-plant
$\tilde{\mathbf{g}}_p$	State feedback controller matrix including neuromotor time constant $\tilde{\mathbf{g}}_p = \begin{bmatrix}\mathbf{g}_p & \frac{1}{\tau_n}\end{bmatrix}$
h	Sampling step size
i	Complex variable. Used also to denote index of an element in a matrix or a vector.
I^k	The skill index at the hierarchy level k
I_i^k	The i^{th} skill index at the hierarchy level k
$\mathbf{I}$	Identity matrix
J	Cost function
J_p	Optimal control cost function in terms of MOCM
k	Index for discrete time or frequency
k_0	Initial gain of HMI
k_n	Gain of HMI
K	Number of executions of a task
$\mathbf{K}$	Solution of the optimal control Riccati equation

$\mathbf{l}_1$	State feedback gain $\mathbf{l}_1 = \begin{bmatrix}\mathbf{l}_p & 0\end{bmatrix}$
$\mathbf{l}_p$	State feedback gain $\mathbf{l}_p = \tau_n \mathbf{g}_p$
$\mathbf{L}_1,\ \mathbf{L}_2,\ \mathbf{L}_3$	Linear matrix inequality constraints
M	Number of possible observation characters
N	Number of samples
N_a	Dimension of $\mathbf{g}_p$
N_d	Order of the Padé approximation
$N_{\mathcal{F}}$	Number of parameters in $\mathcal{F}$
$N_{\mathcal{G}}$	Number of parameters in $\mathcal{G}$
N_s	Number of states in terms of the HMM and in continuous state-space representation
N_y	Number of measurements in $\mathbf{y}$
$\mathbf{N}_0$	Crossterm for optimal control performance index
$\mathbf{N}_1$	Cross covariance for the Kalman filter
o_t	Observation at t
O	$O = o_1 o_2 \dots o_T$ observation sequence within $1 \leq t \leq T$
$p_X(x)$	Probability density distribution
PO	Overshoot percentage
$\mathrm{P_r}(X \leq x)$	Probability given by the cumulative distribution function
PM	Phase margin
q_t	HMM state at t
Q	$Q = q_1 q_2 \dots q_T$ state sequence within $1 \leq t \leq T$
$\mathbf{Q}_0$	State weighting matrix of the optimal control cost function in terms of 0-plant
$\mathbf{Q}_{0a}$	Assumed state weighting matrix of the optimal control cost function in terms of 0-plant
$\mathbf{Q}_y$	Output weighting matrix in the optimal control cost function
r	Control level weighting in the optimal control cost function
R_t	Resource consumption rate
R_{jc}	Jaccard's similarity measure
R_{pr}	Pearson's correlation coefficient
$\mathbf{R}_1,\ \mathbf{R}_{12},\ \mathbf{R}_2$	Weighting matrices of the optimal control performance index in terms of 1-plant
s_ζ	Standard deviation of x around the operating point ζ
S_i	HMM state i
$\mathbf{S}$	Temporary variable for LMI inverse optimal control solution
t	Time. Discrete time index in terms of the HMM. Continuous time in other occasions.
T	Termination time of the observation time range. Discrete in terms of HMM. Continuous in other occasions.
TIME(t)	Global world time at t
u	Input signal of a system
u_c	Commanded control signal in terms of MOCM
u_d	Delayed control signal in terms of MOCM
u_p	Undelayed control signal in terms of MOCM

v_i	HMM observation character i
v_u	Control noise disturbance process (continuous or discrete)
$\mathbf{v}_p$	Noise process for the human operator model states (discrete or continuous)
$\mathbf{v}_y$	Measurement noise process (discrete or continuous)
$V(\theta)$	Performance index with respect to θ
$\bar{\mathbf{V}}(i\omega_k, \theta)$	Frequency function squared error at frequency ω_k
V_u	Control noise intensity
$\tilde{\mathbf{V}}$	Noise covariance matrix of the closed-loop human-machine system
$\mathbf{V}_y$	Covariance matrix of the measurement noise
w_i	The weight of the i^{th} skill index
$w_\lambda,\ w_K,\ w_\Sigma$	Weights used in LMI solution for MOCM gains
$\mathbf{w}$	Plant noise process (discrete or continuous)
$\mathbf{W}$	Covariance of the plant noise
x	Value of a random variable
x_ζ	Value of x standardized around the operating point ζ
$\bar{x}_\zeta$	Average value of x around the operating point ζ
X	Random variable
$\mathbf{x}$	States of a dynamical system
$\mathbf{x}_d$	States of a Padé approximation system
$\mathbf{x}_e$	Difference between the desired states and the true states
$x_{\max}$	Maximum value of x
$\mathbf{x}_p$	States of a human operator model (discrete or continuous)
$\tilde{\mathbf{X}}$	Solution of the Lyapunov equation
$y_e(t),\ \mathbf{y}_e(t)$	Control error $y_r(t) - y(t)$ or $\mathbf{y}_r(t) - \mathbf{y}(t)$
$\mathbf{y}$	Measurements or observations of a system ($\mathbf{y} \in \mathbb{R}^{N_y}$)
$y_r,\ \mathbf{y}_r$	Reference output signal a system
$\mathbf{y}_{\text{obs}}$	Human operator observations in terms of MOCM
$\mathbf{Y}(s)$	Laplace domain representation of $\mathbf{y}(t)$
$\mathbf{Y}(i\omega_k),\ \mathbf{Y}_k$	Frequency domain representation of $\mathbf{y}(t)$ at w_k
$\mathbf{Y}_e(s)$	Laplace domain representation of $\mathbf{y}_e(t)$
$\mathbf{Y}_e(i\omega_k),\ \mathbf{Y}_{e,k}$	Frequency domain representation of $\mathbf{y}_e(t)$ at w_k
$\mathbf{Y}_r(s)$	Laplace domain representation of $\mathbf{y}_r(t)$
$\mathbf{Y}_r(i\omega_k),\ \mathbf{Y}_{r,k}$	Frequency domain representation of $\mathbf{y}_r(t)$ at w_k
$z,\ \phi$	Horizontal position and load angle of the trolley crane simulator model
$\mathbf{Z}$	Inputs for parameter optimization FIS

Greek symbols

α	Parameter adjustment gain in the knowledge-based approach
$\Delta u_p(k+1)$	Increment of the control signal at $k+1$ ($u_p(k+1) - u_p(k)$)
$\delta(t)$	Human control signal at t

$\delta_a(t)$	Assist control signal at t
$\delta_m(t)$	Measured human control signal at t
$\delta_{\max}$	Maximum absolute value of $\delta(t)$
$d\delta_{\max}$	Maximum absolute value of $\dot{\delta}(t)$
$\varepsilon(i\omega_k, \theta),\ \varepsilon_k$	Frequency function error value at frequency ω_k
ζ	Operating point
θ	Parameter vector of the human operator model $\theta \in \mathbb{R}^{N_\theta}$
θ_l	The l^{th} element of θ
$\theta_{\mathcal{F}}$	Parameters of $\mathcal{F}$ ($\theta_{\mathcal{F}} \in \mathbb{R}^{N_{\mathcal{F}}}$)
$\theta_{\mathcal{G}}$	Parameters of $\mathcal{G}$ ($\theta_{\mathcal{G}} \in \mathbb{R}^{N_{\mathcal{G}}}$)
λ	Maximum eigenvalue in GEVP
$\boldsymbol{\lambda}$	$\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ abbreviation for HMM parameters
μ	Location parameter of GEV
ξ	Shape parameter of GEV
$\boldsymbol{\xi}$	Parameters of the HMI
$\boldsymbol{\pi}$	$\boldsymbol{\pi} = \{\pi_i\}$ initial state distribution
σ	Scale parameter of GEV
$\boldsymbol{\Sigma}_1$	Solution of the Kalman filter Riccati equation
τ_n	Human neuromotor time constant in terms of MOCM
τ_p	Dead time delay of the human operator
$\boldsymbol{\Phi},\ \boldsymbol{\Gamma},\ \boldsymbol{\Gamma}_w,\ \mathbf{C},\ \mathbf{D}$	Matrices of a discrete time plant
$\boldsymbol{\Phi}_0,\ \boldsymbol{\Gamma}_0,\ \boldsymbol{\Gamma}_{0w},\ \mathbf{C}_0$	Matrices of a discrete time 0-plant
$\boldsymbol{\Phi}_1,\ \boldsymbol{\Gamma}_1,\ \boldsymbol{\Gamma}_{1w},\ \mathbf{C}_1$	Matrices of a discrete time 1-plant
$\boldsymbol{\Phi}_p,\ \boldsymbol{\Gamma}_p,\ \boldsymbol{\Gamma}_v,\ \mathbf{C}_p$	Matrices of a DDBSFM
$\boldsymbol{\Phi}_p,\ \boldsymbol{\Gamma}_p,\ \boldsymbol{\Gamma}_v,\ \mathbf{H}_p$	Matrices of a discrete time MOCM
$\boldsymbol{\Phi}^{i,j}$	The (i, j) block of a block matrix $\boldsymbol{\Phi}$
$\boldsymbol{\chi}$	States of 0 and 1-plants
$\Psi(s)$	Laplace domain representation of $\delta(t)$
$\Psi(i\omega_k), \Psi_k$	Value of the frequency function $\Psi(s)$ at frequency ω_k
ω	Frequency variable

Other symbols

$\mathcal{F}$	Output nonlinearity in the Hammerstein-Wiener model
$\mathcal{G}$	Input nonlinearity in the Hammerstein-Wiener model
$\mathcal{H}$	General function describing model of the human operator
$\mathcal{O}$	Set of possible observation characters
$\mathcal{S}$	Set of HMM states

1 Introduction

1.1 Background and motivation

Despite the trend of increasing automation degree in control systems the human presence and the human control is still needed in numerous applications. In [139], the following major areas were introduced where the human interactions have a critiral role: aircraft and air traffic control, automobiles and highway systems, trains, ships, spacecrafts, teleoperators, virtual reality for training and entertainment, nuclear power plants and process control, manufacturing of discrete products, hospital systems, command and control in defence and civilian emergencies, office systems, education, smart homes, and personal automation. In addition to the applications mentioned above, the field of mobile working machinery is especially interesting from the perspective of this thesis.

In aeronautics and telesurgery, it is obvious that a high level of human skill is needed for the sake of safety. However, in several industrial processes a high level of human skill is needed as well. On one hand, the operators of these processes need to be skillful so that human error caused accidents would be minimized. On the other hand, in normal operation of industrial processes the quality, productivity, energy efficiency, and abnormal situations (how often they occur and how they can be resolved) depend strongly on the human operator's actions. In chemical industry, according to an old research [110], over 40 % of abnormal situations can be results of human actions. In mining industry, it has been shown in [123] that there are statistically significant performance differences between the work shifts in control of the mineral processing plant. The role of a human operator is emphasized even more in work machine operation such as in forest machines or container cranes. There are research results suggesting that the productivity difference in forest harvester work between two operators working in similar operating conditions with similar machines can be over 40 % [117]. During normal work the difference can be even greater. In addition, the fuel efficiency and quality of the end product is reported to vary significantly between the operators.

Currently the operators of work machines (not to mention the process operators) do not have any possibility to receive detailed feedback of their performance in normal work. However, at least in forest machine work, the operators are very motivated to increase their performance because usually they are paid according to their productivity. Due to the great improvement potential in productivity, quality and fuel economy, there is a growing demand for methods which could help, at least the worst operators to increase their performance closer to the performance of an average operator.

Another trend in industrial processes has been the introduction of high performance onboard computers and processing units with significant amounts of hard disk storage. The machines typically measure a large number of variables which can be stored into a database. For example, the forest harvester performance assessment software Timberlink^{TM} by John Deere Forestry company collects up to roughly 500 variables from processing of one tree. Along with fault detection and maintenance purposes these data can be used to provide the operators feedback of their individual performance. Similarly, the CMS-system provided by the Konecranes company collects a large number of variables with very good time resolution in crane operation. Thus, the measurement and computational capabilities of the machines are typically adequate for

implementation of advanced data processing, control, and analysis algorithms for any purpose.[1]

Consider a human operator performing his/her work using a working machine. In order for the work to be successful and efficient, there are several requirements, which should be met. According to the author's experience, the requirements for efficient operation are:

1. The machine needs to be in good technical condition
2. The human operator needs to know how the work should be done
3. The human operator needs to be able to control the machine efficiently
4. The machine configurations need to be suitable for the human operator's preferences and skill level

All of the requirements are intuitively clear but there are also related research results which indicate their reasonability.

In industrial machines, the first requirement is typically assessed and maintained by the condition monitoring [57, 26], performance assessment [49] and Fault Detection and Identification (FDI) [66], and Fault Tolerant Control (FTC) [66] methods, of which there are wide range of literature available. The second requirement, that is, the knowledge about the work is related to the knowledge and rule-based behaviors in Rasmussen's Skill-Rule-Knowledge model (described in Chapter 5) of the human behavior [127]. In forestry, there are research results showing that expert operators (who also obtained the best performance) could choose a suitable work technique based on the current situation, not only on the preference of using a certain technique [117]. As the human operator learns, he/she makes more successful decisions and correct interpretations of the current state of the system. The third requirement, that is, the ability to control the machine efficiently is related to the skill-based behavior of Rasmussen's Skill-Rule-Knowledge model. While learning, the control actions of the human operator become automatic and integrated and thus tasks can be executed in parallel and overlapping sequences [127]. The fourth requirement, that is, the suitability of the machine control settings has also been researched, for example in the assist control development in [146]. It was found out that the task performance of the human operators increased as the parameters of the virtual internal model were adjusted according to the human operator model.

As described above, in modern working machines the data logging and processing systems provide means to evaluate, assess, and even to optimize all four areas. Because the technical performance evaluation, assessment and optimization is already very well covered in the literature and successful applications exist, this thesis aims at developing methods to the evaluation and optimization of the human skills, work and task performance, as well as adaptation of the machine control settings according to the human skill level.

The role of the human operator varies greatly depending on the application and the level of automation from autonomous systems (no role) and supervisory control systems (supervision and occasional intervention), to manual control systems (authority over everything) [138].

[1] Nevertheless, typically some important measurements (such as exact joint angles in boom cranes or load angle in hoist cranes we well as the operating conditions) are missing because it is difficult or expensive to implement in machines working outdoors in various and harsh operating conditions.

When comparing the extremes, one can notice that the presense of the human operator as a decision maker, a measurement device, an observer, and a controller sets great challenges for performance evaluation and optimization.

The block diagrams of an automatic control system (upper), and a manual control system (lower) are described in Figure 1.1. The dashed lines and boxes describe the quantities or signals which cannot be observed by any (reasonable) means. The automatic (feedback) control system functions generally as follows. The reference signal, which represents the desired states of the controlled machine is fed to the control system. The controller compares the reference to the measured states (obtained from the measurement device), and if they are not equal, a control signal is generated according to a control law (depends on the controller). The control law is introduced to the actuator, which realizes the control action (as force, current, voltage, etc.) and introduces it to the controlled machine.

In the automatic control system, most of the signals can be measured. Of course the controlled machine might have unobservable modes, but using model-based observers, one can at least obtain reliable estimates of them. In the automatic control system, the structures of the individual components are typically known, and their behavior is predictable. In addition, the automatic control system is (typically) time-invariant.[2]

Whereas in the automatic control systems most of the signals and blocks are known and observable, in manual control systems, the most important signals and blocks are often unobservable. In manual control systems (or generally in human-machine systems), the human operator acts as *a controller*, *an actuator*, and *a measurement device (estimator/observer)*. The human observes the states of the system (as well as the possible disturbances) by using his/her senses. It cannot be known which signals the human observes and which senses are used. Whereas the automatic control system uses in servo tasks typically position, and speed signals (whose measurements or estimates can be recorded into a database), the human operator might exploit the vibration or sound of the machine in the control action. The signals the human operator uses cannot be recorded to any database. Moreover, whereas the reference signal (the desired state of the machine) is typically known in automatic control systems, it is, in general, unobservable in manual control systems. The reference signal in manual control systems represents the *intent* of the human operator. That is, a thought of the desired outcome of the control action, which cannot be observed.

When it comes to the "human controller", the control law or the control signal is, in general, unknown. In addition, the human operation is time-variant. The human performance can vary due to emotional factors (motivation, fatigue, focus, etc.), learning, external disturbances, etc. The variance, especially due to the emotional factors is stochastic. Therefore, the human performance is difficult to predict (in short term).

In terms of the performance evaluation and optimization, the unobservability of the reference, measurements used for calculating the control signal, the controller, and the actual control signal is a challenge. The conventional performance measures, such as the Integral of Squared

[2]Here, it must be emphasized that there remains challenges in the automatic control systems as well. The predictability and the controllability of the components might be lost due to physical faults. Moreover, the presense of strong nonlinearities and time variance raise whole new challenges. Nevertheless, assuming flawless technical conditions, the simplification presented here is justified.

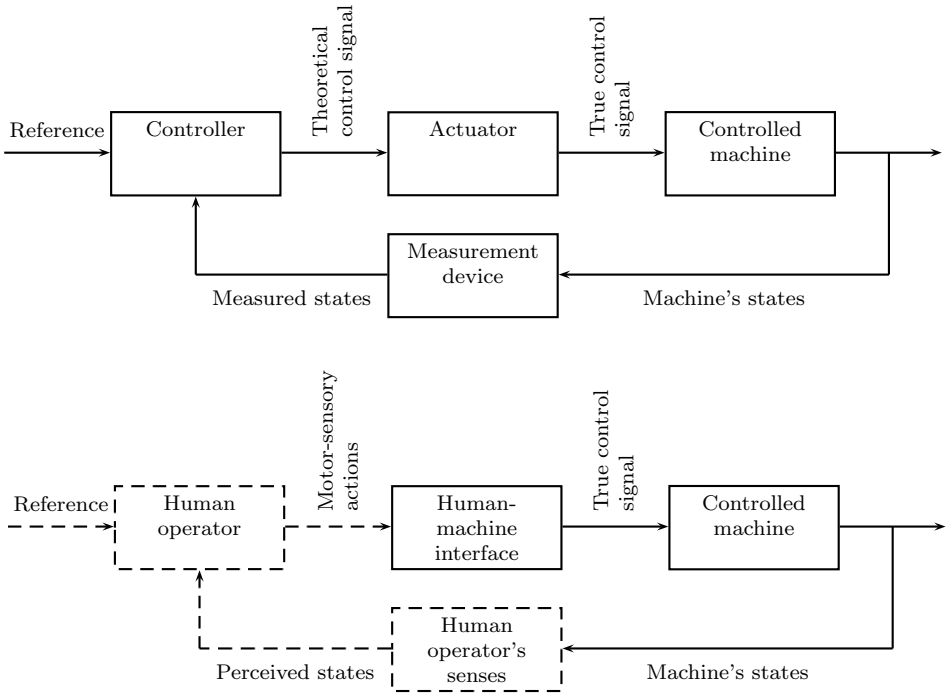


Figure 1.1: The basic structure of an automatic control system (upper) and a manual control system (lower). The dashed lines represents unobservable or unknown signals. Solid lines are observable or known.

Error (ISE) criterion with the control level punishment term

$$\mathrm{ISE} = \int_0^T \Big((y_\mathrm{r}(t) - y(t))^2 + u^2(t) \Big)\, dt \tag{1.1}$$

requires the measurements of the reference $y_\mathrm{r}(t)$, the response $y(t)$, and the control signal $u(t)$. In manual control systems, none of those are generally known. Of course, the response of the system, such as position, might be possible to measure, but the reference signal for the position is not known beforehand because it depends on the intent of the operator. Moreover, the human operator might have additional control objectives in mind, which are unknown and unobservable.

In [47] the challenge of human-machine systems was referred to as information asymmetry. That is, the human operator understands the computer's "way of thinking", but the computers do not understand the human operator (due to the unobservability of the most critical signals) and the computer's inability to understand the human operator's psychological characteristics. Thus, the interaction between the human and the machine is not symmetrical in the conventional human-machine systems. Therefore, the Human Adaptive Mechatronics (HAM) concept was proposed in a Center of Excellence project of Tokyo Denki University in the beginning of this

decade.

The aim of the research in the field of HAM is to facilitate the cooperation between human and machine by developing intelligent tools to measure and evaluate the human operator's skills and help the human operator in task execution. By definition, the HAM system is "an intelligent mechanical system that adapts itself to the user's skill under various environments, assists to improve the user's skill, and assists the human-machine system to achieve the best performance" [145].

In general, a HAM system includes at least some of the following elements [48]:

1. Real-time modeling of human and machine dynamics during the operation, especially under variable constraints.
2. Modeling the skill by using rule, knowledge, and decision combined with event and dynamical systems.
3. Modeling the psycho-physical characteristics of the human operator.
4. A mechatronic system supporting human operation through assisting the control action and providing proper data to facilitate the operator's skill, decision making and performance.

In HAM systems, two fundamental challenges are in understanding, analyzing and modeling the human control action [70], as well as evaluation of human skill in machine operation. Based on the models and the skill evaluation, human adaptive assistance methods can be developed to improve the performance of the human-machine system.

As it is described above, the introduction of condition monitoring and performance assessment systems have enabled the human operators of the mobile working machines to receive detailed feedback and information about the technical performance of the machine. However, as discussed above, especially the high-level performance indicators such as productivity are greatly affected by the human performance. Currently, the operators do not have possibilities to receive detailed feedback about their individual performance or working practices. Moreover, currently the operators do not know whether the control parameters of the machine are suitable for their individual skill level.

In crane work, the swinging of the load (in hoist-type cranes) and the vibration of the boom (in boom-type cranes) is very challenging, because the dynamical properties of the crane change over position and load. Such vibration could be minimized by optimizing the Human-Machine Interface (HMI) parameters of the machine to correspond to the dynamical characteristics and the skill level of the human operator. In addition, the active assist control methods developed in the field of HAM, which modify the dynamics of the human control signal based on the virtual internal model approach and the human model, could even further improve the vibration control.

1.2 Objectives and scope of the thesis

Although theoretical development of HAM reached rather mature state during the Center of Excellence project of Tokyo Denki University, there are still very few practical implementations

of the methods in industrial applications. This thesis studies the methods of HAM from the perspective of mobile working machines. In working machines the operating conditions and operating points of the system (e.g. loads and boom/rope lengths in cranes) vary, and often they can be measured only indirectly *during normal work*. In addition, some of the important variables, such as the load angle or the angles of the joints in crane work, are typically not measured *during normal work*. Moreover, an important application area in the mobile working machines is to provide feedback for the human operators based on their performance, skills and working practices in order to improve the work performance.

In experiments performed under special arrangements, additional measurements can always be installed and used in order to gain all possible information available for development of HAM methods. Most importantly, the objective of the task is known during the experiment. On the contrary, in real-life working machines during normal operation using the measurement data available in the machine, direct measurements of the important variables (such as the load angle in cranes) are not available. In addition, because the human operator can perform several types of work tasks with the machine, the objective of the work is not known beforehand.

The objective of this thesis is to develop and implement HAM methods for working machines, which can be utilized to improve the overall performance of the human-machine system. As a result, a new type of HAM system called Human Adaptive Mechatronics and Coaching (HAMC) is proposed. The system gathers together the structure of a HAM system and the Intelligent Coaching System (ICS) framework, and the corresponding building blocks are introduced from the point of view of mobile working machines. The main blocks of the system are *intent recognition*, *skill evaluation*, *human operator modeling*, as well as *intelligent coaching, assistance and adaptation*. The system contains five interaction levels, which are *performance evaluation*, *skill evaluation*, *coaching*, *assistance (active)*, and *adaptation (passive)*.

The coaching and active assistance are not discussed in detail in this thesis. The coaching level is omitted because the implementation of the system is still ongoing work, and because the author's contribution in the initial development was not as high as in the author's other works described in this thesis. The active assistance is omitted because the theoretical foundation is already very nicely covered in [146, 145, 90]. The reseach work for implementation of the active assistance methods for working machine environment is left for future work.

The first challenge in the development of HAMC is to recognize the *intent* of the operator by using the available measurement data. As a solution, this thesis proposes modeling the work cycle of machine using a Hidden Markov Model (HMM)-based method, which is implemented in three real-life industrial cranes. The work cycle recognition is used to extract the task sequence of the machine based on the sensory data available online in the machine's distributed control system. The HMM was chosen to model the work cycle because it provides a convenient framework for modeling two-level nonstationary stochastic processes (the measurement process and the "task sequence process"), there are readily available software tools for training and using the models, and it can naturally analyze observation sequences of different lengths.

Once the task sequence of the machine has been recognized, that is, the intent[3] of the operator is known, one can evaluate how well the work was performed. This step is called *skill evaluation*. This thesis proposes a skill evaluation method which can be implemented on top of the work

[3]Strictly speaking, the human intent contains more than just the task sequence to be executed. However, this information is enough for skill evaluation, coaching and adaptation of the HMI.

cycle recognition system. The method enables the development of *task specific* skill indices to evaluate the human skill and performance from several perspectives. The skill evaluation serves as a basis for coaching of the operator. The coaching facilitates the human operator's learning process with respect not only to the machine controlling skills but also to the work technique, strategy, and planning. The coaching is performed using the concept of Intelligent Coaching Systems (ICSs) used widely for e.g. training human operators in plants and pilot simulators [25].

Yet another challenge before all interaction levels of the HAMC can be implemented is to identify the model of the human operator as a controller. This problem has been studied for several decades and several solutions have been proposed from the simplest linear Autoregressive with Exogenous Input (ARX) type of models to neural networks and fuzzy systems. From the perspective of working machines, the general use of black-box models (ARX, neural networks, fuzzy systems, etc.) is that they model the human operator's control characteristics in only one operating point.[4] In terms of crane work, for example, the operating point can refer to the mass of the load or the length of the rope the load is connected to.

It is well-known that the human operator can adapt to the changes of the dynamical properties of the controlled system. The black-box models do not have the built-in capability to describe a continuum of human models with respect to various operating points. This challenge can be solved using so called structural models, that is, models whose structures are based on the model of the controlled plant. In particular, Modified Optimal Control Model (MOCM) is a promising method to model the human operators in control tasks. In MOCM, the human operator is modeled as a near-optimal controller and using the optimal control theory, the human models can be adapted to the changes in the machine dynamics.[5] Another tempting property of the MOCM is that it's structure is fixed. As a result, a separate model structure identification procedure is not needed in the modeling procedure, provided that the machine model is of minimal order, so that overlearning due to the over-complexity of the machine model does not occur.

The identification of the MOCM is, however, not an easy task. There exists an excellent iterative procedure to identify the *unknown parameters* of the MOCM based on *known parameters* [32]. However, the set of known parameters includes information such as the motor control noise level, disturbance intensity of the human internal model, the singal-to-noise ratio of the closed-loop response of the human-machine system, the neuromotor time constant, and the effective time delay of the human operator. In addition, the measurements of the responses the human operator is assumed to use in control are needed to obtain the measurement covariance matrices for the Kalman filter. In machine work, most of these parameters cannot be assumed known. Thus, the identification leads to a direct-search-type iterative procedure. In this thesis, the structure of the MOCM is revised so that the unknown parameters can be solved using gradient-based optimization methods even with incomplete measurement information. This is done by relaxing the optimality assumptions and introducing constraints which guarantee the inverse optimality of the feedback gain of the optimal controller and the Kalman filter. Thereafter, the model can be returned to the optimal control framework by solving the inverse

[4] Of course, if the training data set includes samples from several operating points, this information can be exploited in the modeling, but this sounds tedious.

[5] By intuition, this is valid only when the machine's dynamics are not changed drastically. Meaning that if the model should imitate the human controlling a crane, the controlled machine should behave like a crane.

optimal control problem. In addition, it is shown how the known nonlinearities in the input or output of the machine, such as signal level and rate saturation can be taken into account in the human operator modeling using the Hammerstein-Wiener structure. The performances of the developed models are studied in experiments using a trolley crane simulator. The objective of the experiments is to confirm that the developed models perform equally in comparison to the widely used black-box models.

As alternative methods to improve the performance of the human operated machine, *active assistance* and *passive adaptation* can be used. The active assistance, in this thesis, refers to modifying the dynamics of the control signal of the human operator before it is fed to the actual machine. In addition, the active assistance can include a parallel feedback control so that the machine assists the operator in controlling some states (e.g. the load angle swinging in cranes) of the system. The passive adaptation refers to modifying the static parameters of the HMI (e.g. saturation levels or sensitivities in different control directions) to optimize the closed-loop behavior of the human-machine system in terms of some design criteria. This thesis shows that the passive adaptation problem can be returned to a very simple controller tuning problem, where the well-known classical control tuning methods apply. The advantage of the passive adaptation is that since neither the dynamics of the human control signal are modified, nor additional feedback is used, the danger of losing the stability is significantly smaller. In addition, when using the method proposed in this thesis, the adaptation can directly be realized on top of the current control interface in the machine. Alternatively, if the human model is not available, this thesis proposes using the knowledge-based method for HMI adaptation. The method exploits the developed skill evaluation method together with a fuzzy expert system. For implementing the HMI adaptation using data gathered from point-to-point tasks of a crane, a new algorithm is presented. The passive adaptation is realized in an experiment performed using a trolley crane simulator. It is shown that the task performance increases when the sensitivity of the HMI is adapted using the human model or the skill evaluation.

Once the above mentioned challenges are solved, using the developed HAMC, the machine can

- increase the human skill in work, strategy, planning, using the machine, and tuning the machine.
- adapt its Human-Machine Interface (HMI) to be optimal for the operator's *individual skill level and way of working.*
- assist the operator to perform difficult tasks by taking into account the human operator's *individual dynamic characteristics.*

The emphasis in the development of HAMC is to utilize methods which could be implemented in real-life operating conditions for mobile working machines for normal work. Not only in specific test cases and laboratory circumstances.[6] The aim in the experiments is to test whether the methods proposed are useful despite the lacking measurement capabilities in real-life operating conditions.

[6] Although the initial modeling of the human operator presented in this thesis, as well as the passive adaptation was done using a simulator, in the simulator tests the measurement capability of the machine was intentionally restricted. That is, the information of the rope angle of the trolley crane is omitted, even though the human operator surely uses the angle for conrolling the crane.

In [54], the following slogan was presented: "why should people adapt to systems, systems should adapt to people instead?" Indeed, the problem in the use of conventional machines is that the human operator needs to adapt to the machine, not vice versa. In the HAM approach the aim is to ease the role of the human by adapting the machine to the human *individual skill and preferences* as well as *providing help in task execution and working methods.*

1.3 Structure of the thesis

The industrial machines, which are used as case examples in work cycle modeling and skill evaluation are introduced in Chapter 2. The general structure of the HAMC is described in Chapter 3. The chapter also presents the building blocks of the system, as well as describes the possible levels of interaction. Thereafter, the possible application areas of the HAMC from the point of view of forest machines and industrial cranes. Finally, a multi-disciplinary literature survey from the fields related to the development of HAMC is presented.

The HMM-based work cycle modeling method is presented in Chapter 4. Along with the general procedure for the work cycle modeling, the chapter presents a benchmark simulation problem, as well as the results of implementing the method in three industrial case examples. A general procedure for the human operator skill evaluation using the HMM-based work cycle recognition is introduced in Chapter 5. Along with the methodological development, the chapter presents results for the human skill and performance evaluation in three industrial case applications.

The human operator modeling problem (human as a controller) is studied from the structural models' perspective in Chapter 6. The proposes a new structure of Hammerstein-Wiener modified optimal control model of human operator (HWMOCM), as well as a batch method for estimation of the unknown parameters. The concept of data-based state feedback model (DBSFM) is developed. The DBSFM allows to estimate the unknown parameters of optimal control-type human operator models using constrained gradient-based optimization algorithms. This is done by relaxing the optimality assumptions and introducing new constraints, which guarantee the inverse optimality of the identified parameters. It is also shown how the DBSFM can be converted to the optimal control framework by solving the inverse optimal control problem. For estimation of the unknown parameters of the DBSFM, a frequency domain approach is described. At the end of the chapter, a discrete version of DDBSFM (DDBSFM) is developed. The DDBSFM parameter estimation method allows to use the Hammerstein-Wiener approach to account for the nonlinearities in the input and the output of the model.

The experimental results of human operator modeling using a trolley crane simulator are presented in Chapter 7. The experimental setup is described and data gathered from cases of a linear (no saturation or ramps) and a nonlinear (with saturation and ramps) HMI are presented. The identification results of the structural methods described in Chapter 6 are compared to the conventional time-series methods' results.

The structure of a human skill adaptive HMI, used to realize the passive adaptation property of the HAMC is described in Chapter 8. The analytic approach for the HMI adaptation is explained as a controller tuning problem, where the human operator model and the machine model are known, but the HMI is tuned so that the predefined design criteria are fulfilled. Two conventional tuning approaches are described for adapting the HMI using the human model. In addition, a new tuning method, which can be implemented for MOCM-type of models is

presented. A Skill Adaptive Control (SAC) algorithm, designed to realize the HMI adaptation, problem in crane-like machines is described. As an alternative method for solving the HMI adaptation problem, a knowledge-based method is described. The method can be used when the human model is not available. The experimental results using the tuning methods and the SAC algorithm in the trolley crane simulator control task are presented. Finally summary, discussions and conclusions are given in Chapter 9.

1.4 Contributions of the thesis and the author

This thesis focuses on areas of intent recognition, skill evaluation, human model identification, and HMI adaptation. Within those areas, the contributions of the thesis can be listed as follows.

- Description of a system combining Human Adaptive Mechatronicss (HAMs) and Intelligent Coaching Systems (ICSs) called Human Adaptive Mechatronics and Coaching (HAMC) system (Chapter 3)
  - Description of the blocks and the levels of interaction of the proposed HAMC system.
  - Multidisciplinary literature survey about the state of the art in the areas related to the proposed system: plan and intent recognition, skill evaluation, human dynamics identification, intelligent coaching systems, online assistance and cooperative control, and passive human adaptive control interfaces.
- Hidden Markov Model (HMM)-based work cycle modeling for human operated machines (Chapter 4)
  - General framework for recognition of the tasks from human operated work by using HMMs.
  - Implementation of the method for STS cranes.
  - Implementation of the method for RTG cranes.
  - Implementation of the method for overhead cranes.
  - A benchmark simulation for testing the different training algorithms for HMM in work cycle modeling, as well as for comparison of alternative classification methods.
- Skill evaluation method using the work cycle modeling (Chapter 5)
  - General framework for skill evaluation of human operators performing complex work by working machines.
  - Four frameworks for development of skill indices based on task sequence recognition data.
  - Framework for compensation of the effect of external circumstances and operating point from the performance measurements.
  - Implementation and application of the statistical learner model.

  - Framework for aggeragation of the task-based skill indices into skill component evaluations.
  - Development of work performance indices for container cranes.
  - Implementation of the skill and performance evaluation framework for forest harvesters.
  - Implementation of the skill and performance evaluation framework for harbor cranes.

- Identification of human operator's model in machine work (Chapter 6 and Chapter 7)
  - Hammerstein-Wiener formulation of the MOCM (HWMOCM).
  - A batch method for identification of HWMOCM parameters from incomplete measurement data.
  - A novel data-based state feedback model of human operator (DBSFM).
  - Frequency domain gradient-based batch identification method for DBSFM parameters from measurement data which guarantees the existence of the inverse optimal solution.
  - Inverse optimal linear matrix inequality solutions for converting DBSFM to optimal control and estimation framework.
  - A new discrete data-based state feedback model of human operator (DDBSFM).
  - Suboptimal linear matrix inequality solution for converting DDBSFM to optimal control and estimation framework.
  - Hammerstein-Wiener formulation of DDBSFM (HWDDBSFM).
  - Gradient-based time-domain identification method of HWDDBSFM from experimental data.

- Adaptation of the HMI to the human skill and dynamic characteristics (Chapter 8)
  - A general procedure for adapting the HMI using the human model.
  - Formulation of the Skill Adaptive Control (SAC) algorithm applicable especially for crane-like systems.
  - Application of two conventional controller tuning methods for tuning the HMI of the machine: simulation-based tuning and frequency domain tuning.
  - Proposal of a new tuning method in the case when the MOCM-type human operator model is used: Lyapunov-based optimal tuning.
  - A proposal of the knowledge-based method for adapting the HMI using the skill evaluation.
  - An example illustration and implementation of the HMI adaptation as a HMI sentivity design problem.
  - Comparison of the knowledge-based and analytic approaches for the HMI adaptation.

The author proposed the idea of Human Adaptive Mechatronics and Coaching (HAMC) system described in Chapter 3, which includes five interaction levels. The Hidden Markov Model (HMM)-based work cycle modeling method described in Chapter 4 was originally proposed by Lauri Palmroth for modeling the work cycle of forest machines. Simultaneously, the author of this thesis developed an HMM for modeling the stem feeding process of the forest harvester in [155, 156]. The development of the work cycle modeling for the container and overhead cranes was initiated by the author. Although the main work in the work cycle model implementation in the container and overhead crane cases was performed by Samuel Aulanko and Joonas Varso, the development of the general modeling method, training, and validation was done by the author together with Lauri Palmroth. The benchmark simulation for testing the work cycle modeling method, as well as the other possible classification methods was developed solely by the author. The skill evaluation method described in Chapter 5 was developed by the author together with Lauri Palmroth. The statistical learner model described also in Chapter 5 was developed by Lauri Palmroth. The skill evaluation results for the harvester and the container crane applications, as well as the required algorithm implementations presented in this thesis were developed by the author. The human operator modeling results in Chapters 6 and 7 as well as the development of the HMI adaptation system in Chapter 8 are solely the author's results.

2 Description of the cases

In this chapter, the industrial cases used to test the work cycle modeling and skill evaluation methods developed in this thesis are presented. The objective is to provide a comprehension about the working environments of the machines, the tasks of the operator, and the available measurements. In addition, the special experimental arrangements made for the overhead crane case are described shortly. Once the reader understands the basic operating conditions and the work tasks of the machines, it is easier to understand the potential application areas of the HAMC in the work machine domain.

2.1 Mechanized timber harvesting

There are two main methods for mechanical timber harvesting depending on the wood utilization, transportation and the machine types needed. In North America the full-tree method is common whereas in Europe the cut-to-length method is dominant. In the cut-to-length method, two separate machines are used, a forest harvester and a forwarder. A harvester is used for felling, delimbing and cutting trees. Once the harvester has processed the tree stems to logs, a forwarder picks up the logs and carries them to the roadside for further transportation. Pictures of a harvester and a forwarder are shown in Figure 2.1.

2.2 Harvester

2.2.1 Main structure

A harvester can be divided into four main units: engine and power transmission, cabin and controls, boom, and harvester head. A diesel engine rotates the supply pumps which provide the hydraulic pressure for both work hydraulics and hydrostatic transmission systems. The work hydraulics system supplies hydraulic power to the harvester head, to the boom and to all auxiliary functions of the harvester. Hydrostatic transmission works as follows. A variable displacement pump converts mechanical energy from the diesel engine to hydraulic energy. The hydraulics system transmits the hydraulic energy to the wheel axles. A variable displacement motor converts hydraulic energy back to mechanical energy which is mechanically transmitted to the wheels. From the cabin the operator can control the different operations of the machine. The state of the machine and information about processing can be seen on a computer display module. The different parts of the harvester communicate via a Controller Area Network (CAN) bus. [57]

The harvester head is the most complex part of the harvester. Its functioning determines essentially the overall technical performance of the harvester. The main functions of the harvester head are sawing, stem feeding, delimbing of branches, and measuring of the length and diameter of the stem. Figure 2.2 specifies the structure of the harvester head. The stem is felled and cut to logs using a chain saw. The stem is fed to the next cutting point with feeding rollers. The branches are delimbed with delimbing knives which cut the branches due to kinetic energy of the stem and, on the other hand, due to the force from feeding rollers. The harvester head can be equipped with a color marker which marks the logs with color-codes to indicate their

Figure 2.1: A harvester (left) and a forwarder (right).

grades.

2.2.2 Work cycle of harvester and operator's tasks

In order to obtain an understanding about the work tasks of the forest harvester operator and the skills required to perform the work efficiently, the work of the harvester operator in thinning is described in the following. The description is based mainly on the reference [44].

The work cycle and the associated sequence of tasks of the forest harvester operator in thinning is shown in Figure 2.3. It can be seen that some of the tasks are parallel. Generally an expert operator can perform the tasks in as overlapping and parallel sequences as possible, whereas a beginner performs the tasks in series with more cognitive effort. The work cycle and the associated skill requirements can be expressed as follows (adapted from [44]):

1. *Plan the route and drive in the terrain*: The operator's task is to plan the route towards the next working location without damaging the other trees, roots and the environment. The operator moves the machine using a foot pedal or the control lever. The operator can control the balance of the machine by changing the position of the boom, which is operated by two joysticks. This task requires decision making and planning skills as well as silvicultural knowledge. The driving of the machine in uneven terrain and assisting the machine balance by boom requires motor-sensory skills and perception of the machine movements and balance.

2. *Position the machine*: The operator's task is to position the machine so that one can to process as many trees as possible without moving the machine, and to decide the possible log pile positions. Placing the machine is an optimization task which requires years of training until one becomes an expert. The task requires knowledge about the machine reach and capacity, the knowledge and ability to use different work techniques, and spatial pattern recognition skills.

3. *Choose the tree to be removed*: The operator's task is to choose the tree to be removed, plan where to fell and process the tree, and where to place the log piles. This task

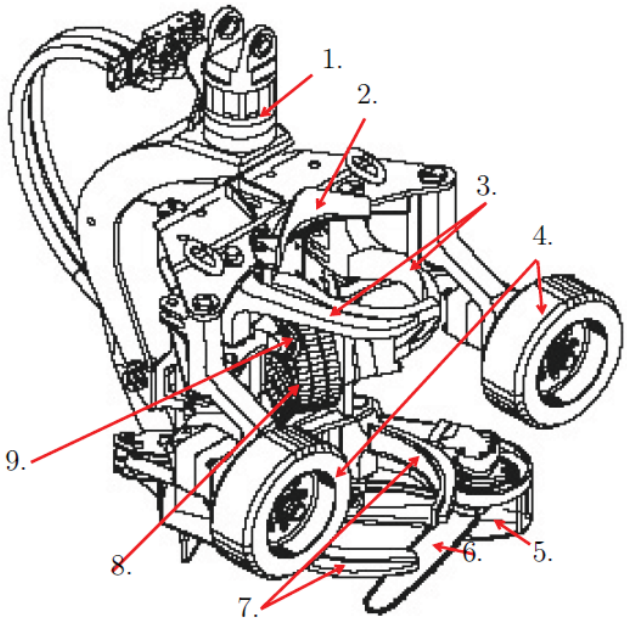


Figure 2.2: Harvester head parts: 1. Rotator, 2. Fixed delimbing knife, 3. Moving delimbing knives (upper), 4. Feeding rollers, 5. Color marker, 6. Chain saw, 7. Moving delimbing knives (lower), 8. Frame rollers, 9. Length measurement wheel. Adapted from [12, 55]

requires knowledge about the work objectives (from the silvicultural perspective), an ability to evaluate the quality of the trees (rot, brances, etc.) as well as decision making and planning skills. Pattern recognition skills are needed to choose the removed trees so that the mutual distances of the remaining trees are suitable (from the silvicultural perspective).

4. *Steer out the boom*: The operator steers the boom towards the tree to be felled. The operator controls the boom using the control levers operated by hands and fingers. The operator should be able to estimate the trajectory and the position of the boom tip, whose speed is about 2 m/s. The boom control requires motor-sensory skills.

5. *Position the harvester head and grip the stem*: The operator's task is to position the harvester head to the tree and to rotate it towards the felling direction. The harvester head is operated using control levers and buttons with hands and fingers. This task requires mainly motor-sensory skills.

6. *Fell the tree*: Fell the tree into the direction decided in the previous tasks. The felling direction should have an open space where the tree can be felled. If suitable space cannot be found nearby, the operator might need to move the standing tree after cutting to the machine trail, which leads to an inverted pendulum type of a balancing task. In the

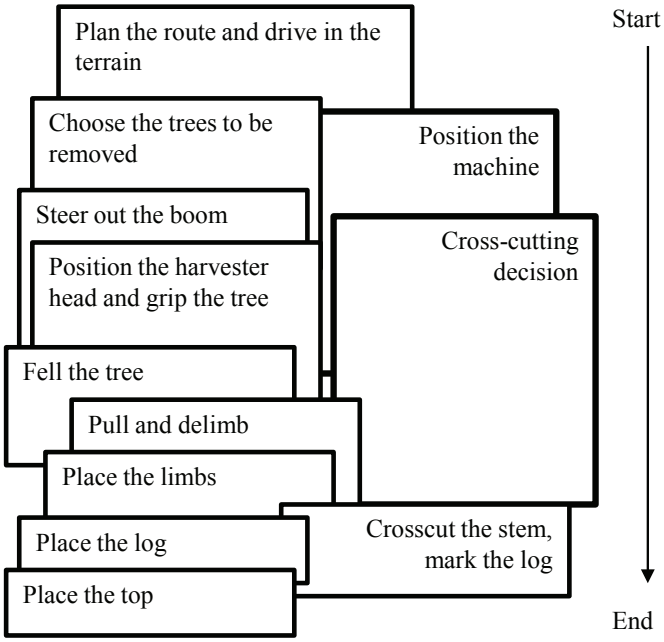


Figure 2.3: The work cycle of a harvester in thinning work. In the task sequence, some of the tasks are parallel. Adapted from [44]

felling task motor-sensory skills are needed to fell the tree safely and avoiding damage to the forest and the machine.

7. *Pull, delimb and place the limbs*: The operator's task is to delimb the branches of the trunk, place the limbs, measure the stem and transport the tree towards the log pile. Simultaneously the operator should avoid damaging the standing trees when feeding the stem using the harvester head. The operator can exploit the energy of the felling tree in feeding and delimbing the first log. While delimbing the tree, the operator examines the quality of the delimbing and, if required, reverses the feeding to complete the operation. The stem speed during the feeding process is about 4 m/s. Thus, the operator needs to make fast decisions regarding the desired length of the tree as well as the delimbing quality. The operator assists the feeding rollers of the harvester head by moving the boom to the opposite direction the stem is fed. Handling of the boom motions during delimbing requires motors-sensory skills.

8. *Cross cutting decision*: The operator's task is to decide the cutting point of the log in order to get the best yield in relation to a price list or a specific order. The operator needs to decide and asses the quality grade of the timber. Only high quality timber is acceptable for sawmills. The lower quality logs are typically used for pulpwood. An expert operator can make the cross-cut decisions for the first two logs already before felling the tree, by visually examining the quality of the standing tree. There is an

automatic optimization system which predicts online the tree profile and suggests the cutting points to optimize the value of the stem, but the final decision is done by the operator. The cross cutting decision requires decision making and planning skills, since the speed of the stem during feeding is fast, and the visual quality examination is difficult during feeding.

9. *Cross-cut, mark the log and spray when needed*: The operator's task is to perform the cross-cut and mark the log grade by color-code using the spray system attached to the harvester head. Typically the cross-cut is done automatically once the operator pushes the corresponding button. However, the operator can assist and stabilize the harvester head and the tree trunk using boom to avoid possible damages to the tree. Using the boom requires motor-sensory skills.

10. *Place the log, limbs and the top*: The task of the operator is to place the logs to piles according to the grade and species, so that the forwarder can later pick the logs easily. In addition, the top remainder of the tree is typically placed on the harvester trail to avoid damage to the terrain. This task requires motor-sensory skills for boom control, planning and decision making skills to avoid waste movements of the tree (the pile positions should have been decided already), and an efficient work technique (how to orient and to which side the log piles are made with respect to the harvester trail).

Generally the beginning of the work cycle involves moving the machine closer to the next tree (if not close enough already), moving the harvester head towards the tree by using the boom crane, catching the tree, and felling it. While felling, the operator assists the felling by the crane. The choice of the next tree depends on the stand and the objectives, but it is the harvester operator who makes that decision. The operator should be able to plan trees to be removed so that the work proceeds efficiently forward, not back and forth. When the operator is catching the tree using the boom crane, the operator should be able to control several joints and functions of the boom and the harvester head using two joysticks and additional buttons. Simultaneously, the operator should have planned the felling location and even the properties of the logs to be processed and the positions of the log piles. While felling the tree, the operator assists the felling by the boom. In addition, the operator can start processing the first log already while the tree is felling.

The tasks 6 - 10 form another work cycle, namely, the stem feeding process. The aim of this subcycle is to feed the stem to the next cutting point to get the best yield out of the stem, either in relation to a price list or a specific order. Harvesters are equipped with a color marking system to be able to separate the various timber grades during transportation. Most of the operations in the feeding process are automated but the operator is still an important factor in the performance of the process. The optimization system which assists in bucking (i.e. cutting the trees into logs) suggests the cutting points so that the value of the stem is maximized. However, the operator has to make the final decisions about the cutting points. The operator can also switch the automatic feeding off, which is typically done in an unexpected situation. In addition, the operator chooses the positions of the log piles by steering the boom to the desired position before cutting the stem into logs.

As it can be understood from the work description above, operating with the forest harvester requires good motor-sensory, decision making and problem solving skills. The operator needs

to be familiar with the control system, be able to make fast and successful decisions, and plan the work such that it is accomplished with minimal amount of non-value-adding motions and actions. The varying operating conditions such as tree sizes, weather and terrain conditions make the operator's work even more challenging. Because the work is very complex, the working techniques between two operators may differ considerably. Each technique has pros and cons. One style can lead to better productivity but the cost can be the loss of fuel efficiency, for example. [158]

Currently the professional operators do not receive any other feedback about their work than the resulting overall productivity. They can get the information about the relative productivity level compared to the statistical reference levels, but not about their work technique or the work tasks in which their performance might be low. The work cycle modeling and skill evaluation methods proposed in this thesis enable to analyze the work performance on task level, and the development of ICS methods to provide feedback for the operator of how to improve the performance.

2.2.3 Data acquisition

The measurements of the harvester head are processed in an embedded processing unit called Harvester Head Module (HHM), which is located on the back of the harvester head. The HHM can communicate with the other harvester processing units via CAN bus. Embedded data processing makes it possible to only transfer a very refined information via CAN bus. Refining the information in the embedded processing units reduces significantly the bandwidth needed in transferring of the data.

In [56], the performance assessment implementation system was divided into three levels: technical level, data level, and application level. The technical level refers to the sensors which transform the physical signals into data. In the data level, the embedded processing and control units of the harvester gather signals, which are then transferred to the CAN bus. The signals include the control lever motions and button presses of the operator, as well as the information from the sensors shown in Figure 2.2. Typically the boom joint angles are not available.

At the data level, there is a data hub, which routes the signals to applications, which can perform online computation and produce different kinds of refined measurement variables. The variables are stored into a database using the Extensible Markup Language (XML), from which they can be exploited by any quality, performance, or skill evaluation software implemented at the application level.

2.3 Forwarder

The task of a forwarder is to load the processed logs from the forest into the load space of the machine, carry them to the roadside and then unload the logs to a pile. In forwarder work the boom operation plays the most significant role in terms of the overall performance of the machine. Unlike in the harvester work, there are no automated functions. Thus, the work of the operator is highly demanding regarding the sensory-motor skill associated with the manual control of the boom. A skilled operator is able to control multiple boom joints simultaneously

and to maneuver the payload fast and precisely, despite that the boom crane has multiple oscillating modes that change significantly over position and load. A work cycle of loading or unloading operation consists mainly of the following basic tasks:

1. Move the grapple to payload
2. Grab the payload
3. Move the payload into the load space or to the pile at roadside
4. Position the payload

A requirement for an efficient forwarder work is to maximize the payload in the load space that is carried to the roadside and to minimize the required driving distance. Usually separate timber grades are transported to separate locations. Sometimes various timber grades may be loaded to the load space which introduces more sorting and arrangement of logs into separate piles.

This thesis focuses on skill evaluation of harvester operators. However, the same methods have been applied for the forwarder operators as well in [118, 159, 119].

2.4 Harbor cranes

2.4.1 Harbor environment

A container harbor is a link between the sea and the land transportation. In the harbor, several different vehicles are needed to fulfill all the required procedures and the tasks. The most important performance measurement is the terminal turnaround time, i.e. how long the ship stays in the terminal. Thus, the objective of the harbor logistics is to maximize the container transportation rate between the ship and the shore. In addition to the efficiency of the individual vehicles, also the cooperation of the vehicles is crucial in order to achieve optimal performance. In general, the planning and control of the harbor logistics can be divided into three levels [163]: strategic level, tactical level and operational level.

There are typically several berths for the ships alongside the container terminal. When a ship arrives to the harbor, the containers are unloaded and loaded by Ship-to-Shore Container Cranes (STSs). Each container ship can be served by several STSs, depending of the size of the ship. In addition, each STS is supported by several other vehicles. Container transportation between the STSs and a container yard can be performed either by Road Trucks (RTs) or Straddle Carriers (SCs). The container yard serves as a storage for the containers and is managed by Rubber Tired Gantry Cranes (RTGs), Rail Mounted Gantry Cranes (RMGs) or SCs. Containers can continue their route from the container yard either to another ship or to the land transportation (by road or by train). Figure 2.4 provides an overview of the container harbor. Containers are colored red, container cranes orange, SCs and RTs black, rail road gate green, and truck gate white. Most of the container harbors use human operated cranes and machines. However, there are also harbors with automated gantry cranes, e.g. Thamesport, Rotterdam and Hamburg [142]. Typically the reasons for automating the procedures are high labor costs or the construction of new a terminal.

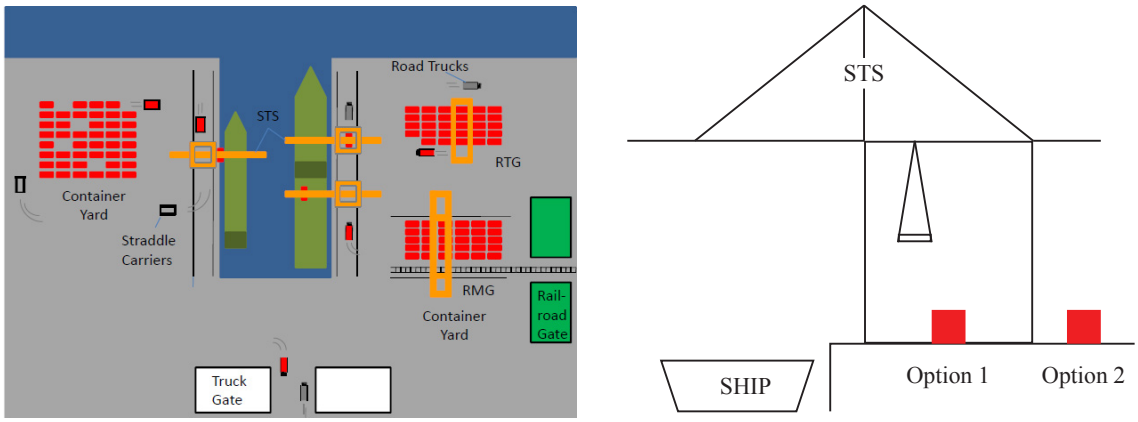


Figure 2.4: An example of the harbor layout and the vehicles (modified from [120]) and the possible container catching points for the STS.

The container yard is usually divided into separate areas, such as import/export stocks, empty stocks, and sheds areas. Container yard includes also the individual sectors for truck and railroad transportation. In addition, certain areas are reserved for special containers, e.g. freezers, and overweight or hazardous containers. Specified areas provide more efficient container transportation at the container yard. Each stock at the container yard consists of blocks of containers, including rows, bays and tiers.

2.4.2 Ship-to-Shore container cranes

The STSs have a crucial role in the container harbor, since they load and unload all the ships. Typically the STS can operate with $13 - 24$ containers wide ships. The trolley of the STS is equipped with a spreader and the trolley travels along the boom of the STS. The spreaders are specified to catch containers, and modern spreaders can lift up to two 20 feet containers simultaneously. The STS has two main tasks: load an unload the ships. Occasionally the STS opens or closes the hatches to the hold, or rearranges the containers in the ship. However, these two task types occur rarely. In this thesis, unloading is referred to as 1 and loading as task type 2. Two possible catching/releasing areas at the shore side of the STS are illustrated on right in Figure 2.4.

Typically the STS moves along the quay on rails. The working power is received as electricity from a power cable. On the top of the STS there is a control room, where the required inverters and other electrical equipment are located. Due to the rails and the electric cable, the STS can operate only in a limited range of the quay.

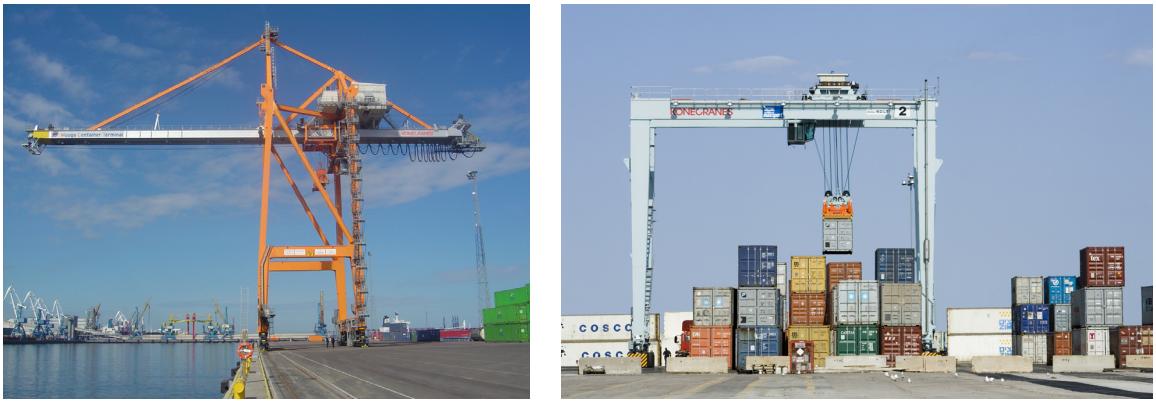


Figure 2.5: Pictures of an STS (left) and an RTG (right).

2.4.3 Rubber tyred gantry cranes

The Rubber Tired Gantry Crane (RTG) works in the container yard. The RTG is very flexible because of rubber tires and a diesel engine. Thus, it is not tied to a certain location in the harbor such as STSs or RMGs. An RTG can commonly stack $8-12$ containers side by side and $4-10$ containers high. The work of the RTG can be categorized in three task types: moving a container from a stack of the container yard to a road truck, from a road truck to a stack, and from a stack to another stack. From truck to stack and from stack to truck task types serve the loading/unloading process of the ships or the land transportation. The last task type aims, for example, at the faster loading of the incoming ships, since the containers are marshaled already into the right order at the container yard. All in all, the RTG has several purposes: to serve the STSs in the loading/unloading process of the ships, to serve the land transportation of the containers, and to rearrange the container yard in order to provide efficient performance later on. In this thesis, from truck to stack task type is referred to as task type 1, from stack to truck as task type 2, and from stack to stack as task type 3.

2.4.4 Data acquisition

Modern cranes have typically sophisticated infrastructure for the data acquisition. The measurement and the computational capacity of the machines are at the level of a regular PC. Several signals can be gathered into databases with high sampling rates. In other words, extensive databases are available for both online and offline analysis of the crane work.

In the cranes studied in this thesis, the data aquisition is done as follows. Programmable Logic Controllers (PLCs) are used to extract raw data from the sensors. PLC can modify and extract some predefined events from the sensor data. In addition, for example the control commands of the human operator can be supervised with the PLC. In order to record the PLC data to a database, signals need to be modified with an I/O server [60]. In the cranes studied in this thesis, the data are stored into an Industrial SQL (InSQL) database. InSQL is based on Relational Database Management System (RDBMS), which enables fast and reliable

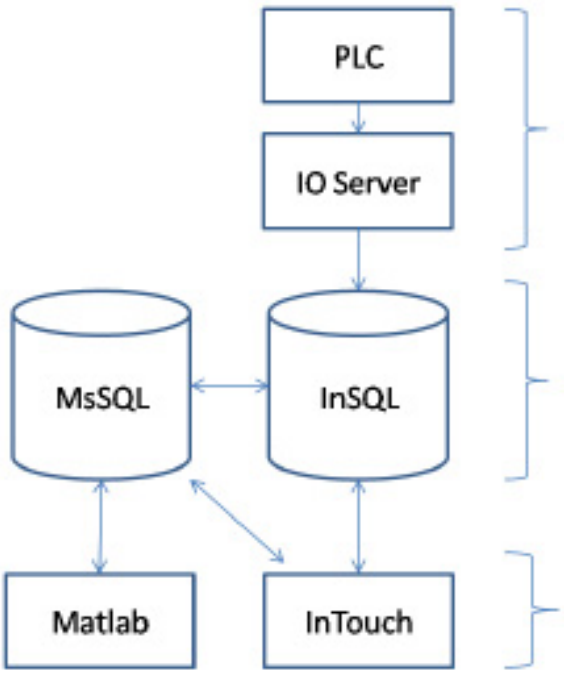


Figure 2.6: A schematic view of the data acquisition of the harbor cranes analyzed in this thesis [120].

performance and high sampling rate. Data can be read from the InSQL directly to InTouch or other applications via Microsoft SQL (MsSQL) server. [59, 143]

In this thesis, Matlab is used for data analysis and development of the algorithms, but in general, any software environment could be used. Figure 2.6 presents an overview of the data acquisition in gantry cranes. In order to provide an understanding about the typical data used as input for the algorithms of the container crane cases of this thesis, the measurements contain the speeds and positions of the different movement directions of the cranes, the operator's control commands, including the use of the spreader corner locks used to catch the container, the information whether the crane is carrying a container, and the mass of the container.

2.5 Overhead cranes

2.5.1 Operating environment

The overhead cranes (or bridge cranes) can work in various types of industrial environments. The cranes can be used in different kind of installation or maintenance work in factories. In such cases, it is typical that the cranes are not continuously operational. However, during the installation work, the crane and the installation staff are typically the bottleneck of the process, because the process under maintenance is typically down during the installation. The faster the installation the lower the downtime.

The overhead cranes can also be used as parts of continuous production processes, such as in steel or paper mills. In such cases, the crane can become a bottleneck due to poor design of the production line (too few cranes) or insufficient capacity (the crane capacity is at least occasionally lower than designed). In the latter case, the causes can be poor technical conditions, poor usability or tuning of the controllers, or an operator with insufficient skill level. The work cycle modeling and skill evaluation framework proposed in this thesis can be used to analyze the causes for the poor performance.

The overhead cranes can be either human operated or automatic. In the latter case, the crane is typically located in a closed environment, for security. In this thesis, it is assumed that the crane is operated by a human operator, but in general, the work cycle modeling method applies for automatically controlled cranes as well.

2.5.2 The Konecranes demo crane

The experiments for the work cycle modeling of the overhead cranes were performed using the Konecranes demo crane. It is a real-life overhead crane whose lifting capacity is 2 tons, lifting height 7.5 m, and span 10.5 m [20]. The crane is primarily used for training operators and testing lifting control technologies. The data acquisition of the crane is similar to those of the harbor cranes described previously. The control parameters of the crane were equal for every operator. The crane is operated with two joysticks.

2.5.3 Tasks and data gathering

Test cycles consisted of transferring a one ton mass between points 1 and 2 as illustrated in Figure 2.7. Between the points the load was disconnected from the crane and the empty hook was moved to the center of the track (point 3). Thus moving with and without mass was included in the work cycle. The distance of the trolley movement needed for this work was roughly 3 m and the distance for the bridge movement 6 m. The load was to be lifted at least to 1.7 m height to pass the obstacles between points 1 and 2. Occasional collisions to the obstacle were not counted. The positions for landing the load in endpoints 1 and 2 were not very strict as the goal width was two times the diameter of the load used. Heights of the points 1 and 2 were the same.

During the work cycles, the mass of the load and position in all three dimensions were recorded into an MsSQL database. The joystick signals were not available in this case but the corresponding speeds were used instead. The data analyses reported in this thesis were performed using Matlab. A video camera recorded an overview of the operators' work during the whole test period. The position of the video camera is shown in Figure 2.7.

2.6 Operator's tasks in harbor and overhead crane work

In order to provide the reader an understanding about the crane work and the operator's challenges, the basic work cycle of typical crane work is described. The work cycle, the operator's tasks and the skill requirements are based on the author's experience and knowledge about

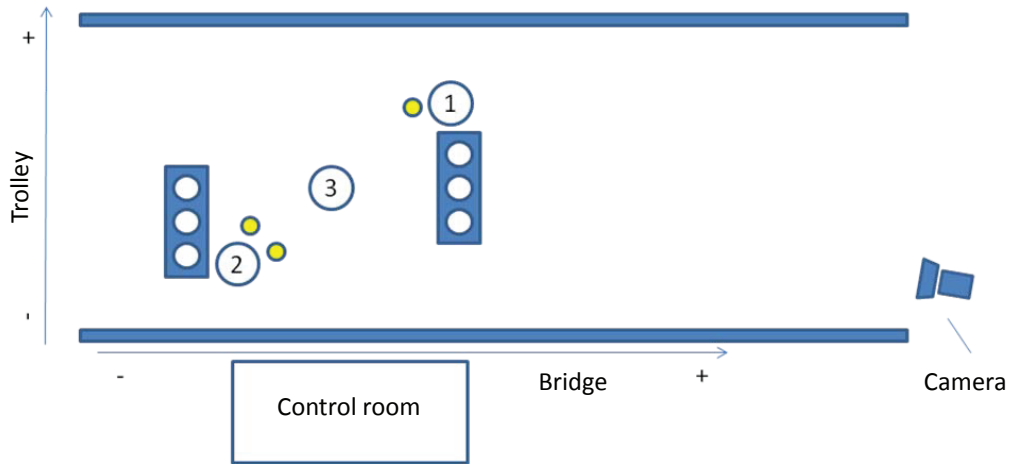


Figure 2.7: A schematic representation of the operator's task in the experimental test arrangements for the demo crane. The picture was originally drawn by Anssi Pekkola.

the crane work after being observed and analyzed the opererator's tasks and work cycles of different cranes in several projects.

1. *Choose the load/container to be transferred*: The operator decides which load/container is transferred next. This task requires planning and decision making skills. Typically in harbor work, the information about the next container is available via the harbor terminal information system. However, especially in RTG work, the suffling of container piles is planned and peformed mostly by the crane operator.

2. *Move the crane towards the load/container*: The operator transfers the crane close to the load/container. An important part of this task is to control the swinging of the empty hook/spreader/grapple. The task also includes planning and realizing the crane motion to avoid collisions with possible obstacles and to obtain a smooth trajectory without unnecessary direction changes or stoppages. This task requires motor-sensory skills as the crane can be moved simultaneosly in three directions.

3. *Lower the crane carefully towards the load/container*: The operator slowly approaches the load/container with the hook/spreader/grapple. The operator needs motor-sensory skills as well as good vision and spatial capacity. In this task, there might be assisting personnel communicating with the operator via a radio or hand signals.

4. *Catch the load/container*: The operator catches the load/container using the empty hook/spreader/grapple. The fine-positioning of the crane is performed in this tasks. Similarly as with the previous task, there can be a person assisting in catching the load/container, especially in cranes where load is connected to the rope via a hook. The skill requirements for this tasks are equal to the previous one.

5. *Lift the load/container*: Lift the load/container carefully from the catching location. The operator sould mind the other people assisting in catching the load/container. Otherwise this is a rather straightforward task.

6. *Transfer the load/container to the desired point*: The operator transfers the crane towards the point the load/container is desired to be released. The operator should keep in mind that the mass of the load/container can easily be tenfold in contrast to the mass of the hook/spreader/grapple. Thus, the operator needs to have an accurate mental model of the behavior of the system with varying loads. In this task, the operator should be able to minimize the swinging of the load. The task also includes planning and realizing the crane motion to avoid collisions with possible obstacles and to obtain a smooth trajectory without unnecessary direction changes or stoppages. Otherwise the skill requirements are equal to the second task.

7. *Lower the load/container towards the releasing location*: The operator slowly approaches the releasing point with the load/container. This is especially difficult in loading containers to the ship, since the containers are placed into cells. It is very difficult to steer the container into the cell that is just wide enough for the biggest containers. Another difficulty occurs when the container should be placed directly onto a truck. This requires good cooperation of the truck driver and the crane operator to ensure that the truck is placed and oriented as well as possible. The operator needs motor-sensory skills as well as good vision and spatial capacity. In this task, there might be assisting personnel communicating with the operator via radio or hand signals.

8. *Release the load/container*: The operator or the possible assisting people release the load/container. The task requires seamless cooperation between the operator and the possible assisting people because fine-positioning of the crane might be needed in order to release the load/container. The skill requirements for this task are equal to the previous one.

Based on the description of harbor and overhead crane operators' work, one can conclude that there are several areas where novel of intelligent solutions could improve the performance and maneuverability of the crane by adapting the machine functions and control settings to the skill level of the operator. In addition, the solutions could help the crane operators to faster learn a proper way of controlling the machine in various operating points, and planning the motion to achive optimal movement trajectories. Moreover, mere task recognition and task performance evaluation during normal work can provide significant amount of useful information for the crane operators as well as the people who plan and manage the logistic chain in which the crane plays an important role.

3 Human adaptive mechatronics and coaching system

Once the reader understands the human operator's work and the related challenges in working machine applications presented in Chapter 2, it becomes obvious that the use of intelligent solutions, such as HAM and ICS methods can potentially improve the performance of the machines significantly. As it was discussed in Section 1.2, the measurement challenges as well as the complexity and the lack of the information of the operator's objective has led to the analysis and development of the HAM methods from the working machines' point of view. This thesis proposes a general framework for the Human Adaptive Mechatronics and Coaching (HAMC) for working machines, which aims at improving the performance by providing feedback for the operator about the his/her current performance and skills, working techniques and potential improvement areas, active assistance in execution of difficult tasks, and passive adaptation of the HMI parameters according to the operator's skill level.

3.1 Overall description

3.1.1 Structure, objectives and assumptions

The main objective of the HAMC is to increase the efficiency of the human-machine combination. The definition of the efficiency depends on the application, but for industrial machines it includes productivity, fuel economy and quality. Thus, one obtains a multi-objective optimization problem with inputs including desired efficiency (the efficiency known to be achievable), the measurements available in the database of the machine, and the current efficiency. The outputs of the system contain information for intelligent coaching, assistance and adaptation.

The structure of the HAMC is shown in Figure 3.1. The structure of the system is similar to the existing HAM systems described e.g. [48]. However, the HAMC described here is designed from the perspective of the mobile working machines, where, as described previously, the measurement capabilities and work complexity set new challenges for the HAM methods. In addition, the HAMC system proposes to use several interaction levels (including coaching), which can be realized depending on the objectives and the available measurement capability. In the following, the HAMC system is described in more detail.

The human operator controls the machine by using the control levers and buttons in the Human-Machine Interface (HMI). In this thesis, it is assumed that the human hand motions or any physiological characteristics, such as heart, brain or muscle activity cannot be measured during normal work of the regular working machines. Currently this is a viable assumption because the industrial machines do not include any sensors which could be used to measure those variables. Instead, it is assumed that the control lever motions and the button presses are recorded in the machine's database. In the figure these are described as human control commands. It is assumed that the HMI is parametrized so that the properties of the machine can be set to correspond to the operator's skill level. The parameter setup might include for example the sensitivity of the movements, the maximum allowable speed for different movements, and the configuration of the rate limiters (ramps) designed to smoothen the accelerations of the controlled elements.

Furthermore, it is assumed that at least some of the states of the machine are measurable,

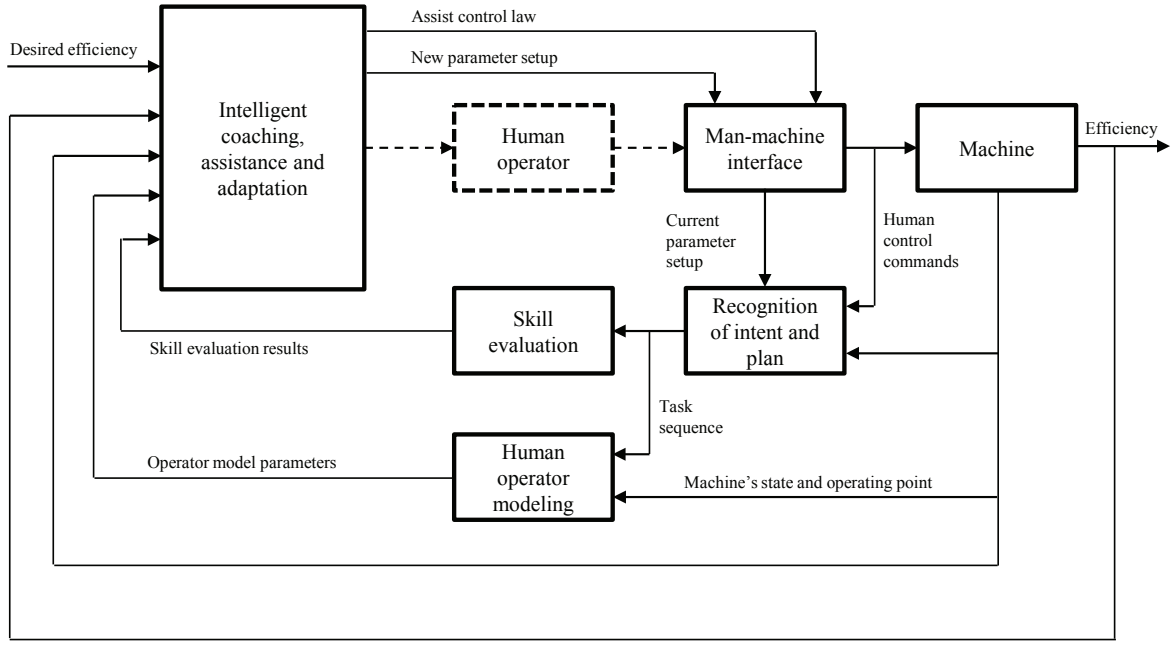


Figure 3.1: The block diagram describing the general structure of the HAMC system aiming at increasing the efficiency of the human-machine combination and maneuverability of the machine.

or can be estimated. The states here refer as well to the discrete states of different machine functions as to the continuous dynamical states of the controlled element. It is also assumed that the operating point of the machine can be measured or estimated. The operating point includes the physical variables which introduce natural constraints to the performance. For example, the mass and size of the controlled element affects to the performance in machine work. Moreover, especially in crane operation the performance is greatly affected by the physical distance (horizontal and vertical) the object is transferred.

In machine work and generally in human-machine systems, a significant challenge is to determine the intent of the human operator. The intent cannot be measured. It can only be estimated indirectly from the observations available. In this thesis, the intent recognition refers to the classification of the task sequence hidden behind the measurable observations. The difficulty and reliability of this estimation task depends on the available sensors and measurements. In addition, it is very difficult to recognize the intent online, because the working machines can typically perform several different types of tasks and sequences. There are successful examples of estimating the intent online, but in machine work with rather modest amount of available measurements, this is still quite a challenge. For coaching and passive adaptation it is enough to estimate the intent and plan after the task sequence is executed. However, for direct assistance, the online intent recognition is needed.

Once the intent of the operator is known it is possible to estimate the performance and skills of

the operator. One could describe this as "now that we known what the operator was trying to do we want to known how well he/she did it". In this thesis this is called skill evaluation. In the skill evaluation the operator's performance in different tasks is evaluated with respect to the overall performance, skills in machine control, machine tuning[7], planning and decision making, and work technique and strategy. The skill evaluation can be used as a basis for intelligent coaching of the human operator as well as adaptation of the HMI.

In addition to the skill evaluation, the measurements of the machine states and the human operator's control commands can be used to model the human operator. In addition, the human operator modeling can exploit the information of the task sequence given by the intent recognition. The objective of the human operator modeling is to find a dynamical model which can predict and simulate the behavior of the human operator as a controller based on the measurements available in the machine. The model can be used to determine the optimal parameter configurations for the HMI and to determine the parallel control law to assist the operator to perform some difficult tasks.

In Figure 3.1, the block "intelligent coaching, assistance and adaptation" contains the algorithms and methods to fulfill the objectives for coaching, assistance and adaptation. The block contains several outputs. Firstly, it provides information and feedback to the operator to increase his/her performance. The feedback is generated using the framework of ICSs. The feedback can be presented in different forms, such as linguistic instructions (as text or sound) or pictographic multimedia presentations (in the machine's display). If it is possible to recognize the intent of the operator online, the machine can provide direct assistance for the operator to perform the task. In some tasks of the machine work, the intention recognition can be done rather easily. An example of this is the load swinging stabilization in the crane work. In such tasks, the machine can assist the operator to compensate the swinging of the load. In practice, one can introduce a parallel automatic controller which can complete the human control signal so that the objective of the task (e.g. swinging compensation) would be fulfilled optimally. Alternatively, the controller can override or complete the human operator's controls in these tasks.

The last phase of the "intelligent coaching, assistance and adaptation" block is the adaptation. In this phase, the objective is to find the parameters of the HMI which would lead to best performance, given the current skill level. This is an optimization problem which can be solved analytically or heuristically. The analytic solution requires the models of the human operator and the machine, measurements available in the machine, and the information of the current parameter setup. The heuristic solution uses the information of the current parameter setup and the results of the skill evaluation as inputs to an inference system which determines whether the parameters should be modified.

To summarize, the steps which need to be taken in development of the HAMC are:

1. *Intent recognition*: Determine the task sequence of the human operator by using the available measurements.

2. *Skill evaluation*: Evaluate the overall task performance, skills in machine control, machine tuning, planning and decision making, and work technique and strategy.

[7]Here the machine tuning skills refer to the operator's ability to choose the control settings of the machine appropriate to his/her skill level.

3. *Human operator modeling*: Identify the human operator model based on the available measurements.

4. *Performance improvement*: Based on the skill evaluation and/or the human operator model, determine how the performance can be improved.

Depending on the *levels of interaction* designed for the HAMC, the human operator modeling or alternatively the skill evaluation might not be needed. The possible levels of interaction for the HAMC are discussed in the following.

3.1.2 Levels of interaction

The proposed HAMC framework can interact with the human operator on different levels and modes. Although being an important part of the HAMC, especially in the coaching phase the modes of interaction, such as the form the feedback is presented for the operator, are not discussed in this thesis further.

The levels of interaction between the HAMC and the human operator are defined as:

1. *Performance evaluation*: The system evaluates the performance of the human-machine system on the production level. The productivity evaluations along with the performance of some machine functions are presented for the operator. Providing statistical reference values regarding the key performance indices can be enough for a motivated operator to increase his/her performance.

2. *Skill evaluation*: The system evaluates the human operator's skills from different aspects. The system provides the evaluation results for the operator but does not give detailed instructions of how to improve the skill. The skill evaluation enables the operator to see his/her most important improvement areas.

3. *Coaching*: Based on the skill evaluation the system analyzes the work and skills of the human operator in different tasks. The system can give task specific instructions of how the performance could be increased. The detailed instructions facilitate the learning process of the operator in performing complicated work tasks which require good problem solving skills.

4. *Assistance (active)*: In addition to coaching, the machine can assist the operator to perform certain difficult tasks. The assistance increases the performance of the novice operators in tasks which require high motor-sensory skills.

5. *Adaptation (passive)*: If the HAMC recognizes that some parameters of the HMI are not suitable for the operator's current skill, the system can automatically change these parameters to improve the maneuverability of the machine. Adaptation increases the performance of the novice operators in tasks which require high motor-sensory skills. In addition, the adaptation can increase the performance of the skilled operators e.g. in a case where the current parameters are tuned to for a novice operator.

This thesis focuses mainly on the evaluation and adaptation, that is, the first, the second and the fifth level, but also the other levels can be achieved using the methods proposed

in this thesis. The realization of the third and the fourth levels are left for further work. The coaching level is omitted because the work reported in [119], of which the author of this thesis was also part of, was mainly done by the first author of the paper. In addition, when writing this thesis, the experimental test results of the coaching were not available. Thus, the coaching could be only reported in a conceptual level without long-term follow-up about the performance improvement effects. The implementation of the coaching part is ongoing work. However, already the passive adaptation results described in Chapter 8 can be used as coaching instructions for the operator, if the parameters are not adapted automatically. This is also a form of coaching in the *parameter tuning* coaching problem.

The active assistance level is omitted because the theoretical foundation is already very nicely covered in [146, 145, 90, 42]. Although the implementation of the method in the work machine domain might require additional research due to the lacking measurement capability, the uncertainty about the operator's intent, and the use of the virtual internal model control, the research already done on assist control provides a solid base to implement the methods in working machines as well.

3.1.3 Relations to the Norman's model of human action

Although the HAMC is not a model of the human operator, but it describes a system structure which can be used to increase the performance of the man-machine system, it is interesting to compare it to the human's action and evaluation strategies. It turns out that the HAMC basically performs a reverse process of the human action cycle. A well-known model of the human action in activities for everyday life as well as for work tasks is the Norman's seven stage model [112]. In the model, the human action is divided into processes of forming a goal, execution of the action to achieve the goal (gulf of execution), and the evaluation of the result (gulf of evaluation). According to the model, the seven stages of human action can be expressed as:

1. *Forming the goal*: Determine the goal of the work (can be imprecise at this stage).
2. *Forming the intention*: Refine the goal into a more precise decision describing how to achieve the goal and what activities needs to be performed achieve the goal.
3. *Specifying an action sequence*: Transform the intetion to the specific tasks and the sequence that need to be performed to achieve the goal.
4. *Executing an action*: Perform the tasks in the specified sequence to achieve the goal.
5. *Perceiving the state of the world*: Observe the output of the actions in the world.
6. *Interpreting the state of the world*: Intepret the output using the knowledge about the world to understand the result of the actions in the world.
7. *Evaluation of the result*: Compare the result to the desired goal.

The aim of the HAMC is to improve the performance of the man-machine system by providing feedback about the current performance with respect to a statistical reference and skill level

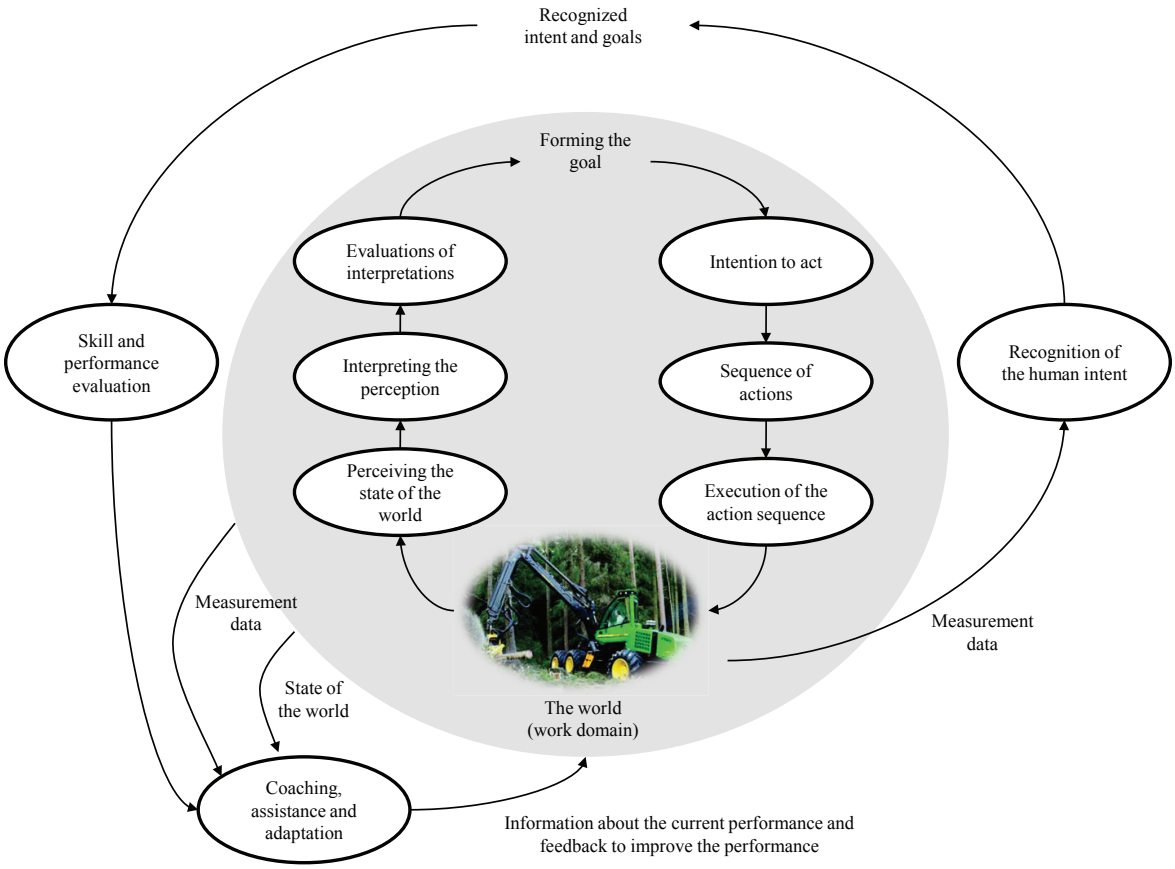


Figure 3.2: The Norman's model of human action cycle and the relation with the blocks of the proposed HAMC.

of the human operator, as well as to assist the operator in difficult tasks (e.g. load swing or vibration of the boom in machine work), and to adapt the system parameters to make the controlling of the machine easier. The cycle of the HAMC is illustrated outside the human action cycle in Figure 3.2.

In order to achieve the performance improvement, the HAMC needs to recognize the intent and the goal of the human operator. This can be thought as a reverse process for the "gulf of execution" in the Norman's model. That is, based on the measurement data available, try to infer "which tasks were executed in what order and what was the final goal of the work?". Once the tasks and the task sequence has been recognized, one needs to evaluate how well the work was performed. That is, "was the goal achieved?" and "how well the tasks were performed?". This, on the other hand can be thought as the reverse process for the "gulf of evaluation". Based on the skill and performance evaluation, the system can analyze "what could have been done better?", "were the machine parameters suitable for the operator?", or simply present the performance for the operator to help him/her interpreting the results of the actions with

respect to the world. In terms of the Norman's model, the aim of the HAMC is to "improve the state of the world with respect of the human operator performance".

3.2 Applications of HAMC in working machines

The objective of this thesis is to develop practices to utilize HAM and ICS methods in working machine environment under the framework of HAMC. Therefore, in the following the potential applications of the HAMC in the work machine domain are outlined.

1. Provide task-specific performance data for the operator and the work management with respect to the statistical baseline. When the operator is given task-specific information about his/her performance, he/she might be able to figure out how to improve the performance. In addition, providing the work planners and managers information about the performance with respect to the statistical baseline, the managers obtain new tools which can be used in decision making related to the work chain configuration (how many and which machines are in the field) and the potential bottlenecks in the work chain.
2. Provide detailed instructions how to improve the performance in certain restricted coaching problems. Such coaching problems can be for example:
   (a) Assisting in choosing a suitable grapple size in forwarder work or work with grapple cranes.
   (b) Assisting in choosing the correct machine location in forwarder and harvester work. In both cases the machine should be located so that one can catch more than one tree/pile without driving forward/backward.
   (c) Assisting in planning the harvester and forwarder work so that the backward driving is mininized.
   (d) Teaching systematic working procedures for the harvester/forwarder operator.
   (e) Assisting the operator to choose the machine control settings suitable to his/her skill level.
   (f) Assisting the harvester operator to choose the pile locations so that the total performance of the harvester-forwarder work chain is optimal.
3. Provide active assistance to minimize the vibration of the boom in harvester/forwarder work or minimize the load swinging in crane work. Especially for novice operators the payload swinging in the crane work is challenging.
4. Adapt the machine control settings automatically to correspond to the operator's skill level. If the control settings of the machine (the sensitivity in different movements, ramps (rate limiters), etc.) are not set to correspond to the human operator's skill level, the crane control is difficult. The payload might start swinging and the compensation of the swinging might be difficult with too sensitive/fast control settings. This decreases the performance of the work and increases the danger of hazards.

The HAMC methods described in this thesis can be used to develop new type of training simulators or even coaching systems which can be utilized to coach the operators during normal

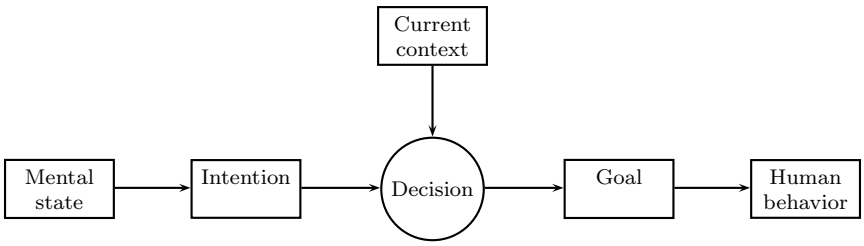


Figure 3.3: Generation of a human behavior according to [172].

work. In addition to training the novice operator, the methods might help the experienced operators to further improve their performance since they get a change to reflect their own performance against a statistical reference. Moreover, the the maneuverability of the system can possibly be improved by the using control settings and assistance designed individually for the current operator, instead of using the general settings.

3.3 Building blocks and the state of the art

3.3.1 Plan and intent recognition

Recognition and estimation of the human intent or plan is a very important research topic in areas such as human-robot cooperation, human-machine collaborative systems, and teleoperation. Often in the literature no distinction is made between the intention recognition and the plan recognition. Therefore, in the following review, both areas are considered. However, in some cases a clear distinction is made, such as in [151] the plan is defined to be composed of a sequence of actions to achieve the desired goal, and intention is defined as the explanation behind a specific action. Because in practical applications the human operator's intent recognition needs to be based on incomplete observations from noisy sensor data, this review focuses mostly on probabilistic methods for plan and intent recognition.

In [172] the human intention is explained by introducing the process of human behavior generation shown in Figure 3.3. According to [172] the intention represents and idea or mental state of what the human is going to do. Based on the intention, the human creates a goal and plans a sequence of actions which lead to that goal. The intention recognition can be seen as a reverse process of the human behavior generation. Based on the observable actions, the human's goal can be recognized. By putting the goal and the sequence of actions into a context, the intent of the operator can be recognized. In [172] the intention recognition is carried out by defining an "intention graph" whose hierarchy is based on the reverse process of human behavior generation described above.

The human intention is not observable, but some of the actions can be observed. Most intention recognition systems described in the literature seem to exploit that natural hierarchy. In addition to the hierarchical structure, due to the uncertainty inherent in human action and decision making, the human behavior is stochastic. Therefore, deterministic models are not suitable for human behavior modeling. Dynamic Bayesian Network (DBN) provides a general

framework for modeling partially observable hierarchical probabilistic systems. In [150] a DBN is proposed for human intent recognition in human-robot interaction. The network structure proposed in [150] is based on the description of three level intentional behavior of an intentional agent in [51]. The levels of intentional behavior are described as

1. The intentional level: agent's beliefs, goals, plans and desires.

2. The activity level: agent's tasks, functions and actions.

3. The state level: observations of the agent's behavior.

In the network structure proposed in [150], the top layer represents the intentions and the actions constructed at the second layer. The bottom layer is set up from the observations. The temporal dynamics in the "intention-activity-state" process is modeled by using the DBN representation. The intent recognition method is utilized to achieve compliant human-machine interaction in [151, 149].

Dynamic Bayesian networks are also utilized in [135], where the objective is to develop a generic human intention recognition system for human-robot cooperation. It is argued that the intention recognition system should be able to handle both continuous and discrete observations. Thus, a Hybrid Dynamic Bayesian Network (HDBN) structure is proposed to model the human intentions based on the observations from the process. Because the human intentions can be affected by the external circumstances, it is also taken into account. The external circumstances are referred as "domain knowledge", which needs to be defined separately in the modeling process. The model is applied in proactive human-robot cooperation in [137]. Moreover, a method which can be used for reducing the state-space of large Bayesian networks to provide tractable models for human intention recognition is proposed in [136]. In [86], the DBN is utilized for intention recognition for partial-order plans in assembly work.

An HMM is a type of DBN which has been widely applied in human action modeling and human intention recognition. In [174] HMM is used for motion intention recognition in telemanipulation. The objective is to provide virtual fixture assistance for enhancing the performance of teleoperation. Windowed Fast Fourier Tranformations (FFTs) of the measurement data quantized by using vector quantization are used as inputs for the HMM. The model classifies the motions correctly with the accuracy of around 90 %. In [152], the human motions are recognized using separately trained HMMs with the aim at improving the nonverbal communication between human and robot. The motions recognized by the HMMs are used together with HMMs at a higher level to recognize suitable interaction patterns for the robot based on the human motions. In [22], the HMMs are applied for robotics. The objective is to to realize the concept of *learning by imitation* for a robot. The HMMs are used to recognize the human gestures after which the robot determines the optimal trajectory for the imitated task by taking into account the possible task constraints. The final trajectory tracking is realized by an optimal controller. Similarly, the learning by imitation concept is utilized in [87], where online learning method for motion primitives is proposed. The observations are segmented and clustered by a hierarchical method to represent the known motion primitives. The motion primitives are then encoded using HMMs. The same HMMs are also used for recognition of the human motions as well as generation of the robot's motions. In [88] the method is extended

for Factorial Hidden Markov Models (FHMMs), where multiple dynamic stochastic processes are allowed to interact, in contrast to a single two-level stochastic process in a regular HMM.

The HMM-based intention recognition method is developed further in [2], where a Layered Hidden Markov Model (LHMM) is proposed to model the human intentions. The LHMM consists of the gestem level and the task level. The gestem level classification results act as inputs to the task level model, which provides the final task recognition results. The aim of the LHMM-based intention recognition is to implement the adaptive virtual fixture method which assists the human operator in computer-aided machine control in [1, 36]. The HMMs have also been used to infer the intent of the human in multi-modal robot programming [61] and to estimate the intended dancing steps for dance partner robot [153]. A Hybrid Hidden Markov Model (HHMM) is utilized to model and recognize the human intentions for human-robot interaction in [58]. In the HHMM each hidden state consists of a dynamical system. The state sequence can be estimated by using the hybrid Markov observer.

Neural networks and fuzzy reasoning have also been used to recognize human intenions and plans. The use of neural networks in recognition and modeling the human actions in human computer interface applications is discussed already in [38] in a general level. Neural networks and fuzzy sets are used in [46] to recognize the intentions of a tank driver and predict it's low-level actions. Moreover, the intention recognition using conceptual fuzzy sets is proposed in [64]. The objective is to improve the communication between the user and the computer by recognizing the intentions of the user based on his/her actions. The fuzzy reasoning is used in [65] to recognize the human intentions with the aim at improving the nonverbal cooperation between the human and the robot. In [77], the human gestures are recognized by using fuzzy reasoning, where the input data are obtained using a wireless data glove system. Based on the recognized gestures the computer can react to the commands of the user.

To model the intentions and plans of a human operator in machine work, the HMM-based work cycle recognition method is decribed in [118]. Simultaneously, the author of this thesis a implemented similar method to model and evaluate the stem feeding process of a forest harvester in [155, 156]. The stem feeding process is partially automated, because the human can also choose to operate the process manually. In addition, the human operator has the authority over the final decisions. Most importantly, the stem feeding process can be divided into separate tasks, which can be recognized by the HMM using the process measurements as inputs. This thesis describes a general framework for utilizing the work cycle modeling method for industrial machines.

For neural networks, fuzzy reasoning, and HMMs there exists readily available and widely used official toolboxes for training, analysis and use of the methods. There is a good toolbox for the Bayesian networks as well. Since the objective in the work cycle recognition is to classify the dynamic time-series data into sequences of tasks (performed by the operator), the dynamic models such as DBNs and HMMs are natural choices. In fact, an HMM is only a special case of a DBN.

The HMMs were chosen for human intention recognition in this thesis because they provide natural means of modeling and classification of temporal context-dependent sequential stochastic signals using probabilistic methods. In addition, there are widely accepted ready-made tools available for training and using the models, and the algorithms are very easy to implement in realtime environment with any programming language. An HMM is also directly applicable

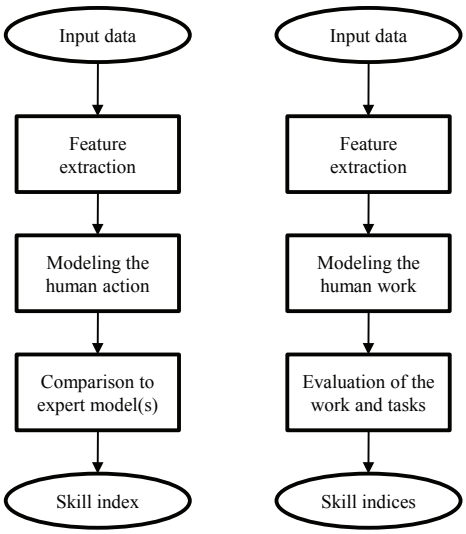


Figure 3.4: Different approaches for automatic skill evaluation. On the left, the model matching approach and on the right, the work modeling approach.

for observation sequences with different lengths, and using the single HMM with states corresponding the work tasks the computational cost of the task sequence estimation is very low. The HMM-based work cycle recognition method is described in Chapter 4.

3.3.2 Skill evaluation of machine operators

The application areas of the human skill evaluation are widespread. The skill evaluation has been used at least in robotics for skill transfrerring from human to robot or from robot to human [16, 126, 97], medical engineering and surgery [100], commercial and military flight systems [166], and nuclear power plant simulators [78]. Most of the above mentioned applications require high level of human skill in order to guarantee safe and efficient operation. The skill evaluation is also a prerequisite for development of ICSs or HAM systems. In the following, a short literature review of the skill evaluation of human operator in machine work, or machine/robot assisted work is given. The review focuses mainly on the application side of the skill evaluation, not on the cognitive science perspective. Moreover, the conventional methods for skill evaluation which are based on separate tests and recordings are out of the scope of this thesis. Only the methods used for automatic skill evaluation are considered.

The Fitts' law is the basic concept for evaluation of human skills in execution of peg-in-a-hole type of insertion task [39, 82]. The Fitts' law describes the relationship between a task performance metric and task difficulty index, which is defined by the ratio between the movement distance (from the beginning to the center of the target) and the width of the target. In [82] an extension to the Fitts' law is proposed. As a result, the concept of Fitts' law can be used to evaluate skills decribed by more than one variable. In machine work however, distances, sizes and positions of the targets vary and often they cannot be measured.

The automatic skill evaluation systems found in the literature can be divided roughly to two classes: the model matching approach and the work modeling approach (Figure 3.4). In the model matching approach the human operator's action is modeled as a whole based on the features extracted from the measured data. The model (or observations) from the current human operator are compared to the model of an expert operator (or expert operators). The resulting skill index describes how well the actions of the current operator match to the actions of the expert operators. To realize the model matching approach, a human operator model is needed. Parametric or nonparametric control theoretic models can be used to model the human operator actions (as a controller) in execution of a control task. The human skill quantification can then be based on the parameters of the identified controller, time or frequency characteristics of the control signal, or on comparison of the expert model control trajectory and the realized trajectory [89, 133, 52].

Another method used to utilize the model matching approach is to train dynamic probabilistic models (such as HMMs) to represent different skill levels. The evaluation is based on the posterior probability of the models of different skill levels (given the input data), or on the likelihood index (the likelihood of the data given the model) if there is only one model against which the data are evaluated. This approach has been utilized at least in [167, 173, 100].

Fuzzy Inference Systems (FISs) have been used for skill evaluation of crane operators in [168] in order to realize an interactive adaptive interface. The input data for skill evaluation are the difference between the proper control pattern for load swing supression and the human operator's control pattern, as well as the operation time for swing supression. The proper control pattern could be defined mathematically based on the hoist angle measurement. The method used in [168] falls basically under the model matching approach, although the use of performance metrics such as operation time is more typical for the work modeling approach.

In addition to the FIS, also Self Organizing Maps (SOM) have been used for skill evaluation in [148]. In particular, the task scheduling of the human operation was studied in a multi-task environment consisting of radio-controlled construction machines. The resulting SOM-based method could successfully visualize the task switching strategy and the learning of the human operator.

The work modeling approach is usually carried out by using HMMs to separate and recognize the tasks of the work based on the observed data. In the framework of this thesis, the task sequence recognition is called the intent recognition, or the work cycle recognition. In [106], separate HMMs are trained for each motion of the surgeon's work, and the motions constituted a network of HMMs. The skill evaluation is based on motion frequencies and the time used for different motions. In [132, 131], the surgeons' performances are evaluated by defining the work of the surgeon as task sequences, which are modeled by using HMMs. The skill evaluation is based on a statistical distance to an expert model, task execution times, task execution frequencies and force/torque measurements during tasks.

This thesis presents a general framework for analyzing the performance and skills of the human operators in machine work by using the HMM-based work cycle modeling approach. This method is chosen because it enables to analyze the human work consisting of sequential work tasks at *task level*. It is shown in this thesis that using the work cycle modeling approach one can develop task-specific skill indices to measure different aspects of human skill and performance. In other words, the interpretation of the skill and performance indices developed using the

proposed framework is significantly easier than e.g. matching an observation sequence to an abstract probabilistic or control theoretic expert model using some similarity measure. The skill evaluation method using the HMM-based work cycle recognition is described in Chapter 5.

3.3.3 Human model identification

The dynamic behavior of human operators has been studied for decades. Depending on the use of the human operator model, various model structures and methods have been used. Examples of model structures are quasi-linear models, crossover models, optimal control models, different types of linear parametric and time series models, different types of nonlinear models, as well as adaptive and knowledge-based models. In the following, only the parametric models of human operator are discussed. Basically the structures can be divided into three categories: linear models (structure based on identification theory or assumptions of the human dynamics), linear structural models (structure based on the model of the controlled system), and black-box models.

In the quasi-linear model the human operator is described with a linear differential equation and a nonlinear part for noise and other disturbances in the system. The nonlinear part is called a remnant and is described by statistical quantities [99]. In [105] the quasi-linear multi-loop model is used for car-following identification. Although the identified parameters varied among the subjects, the models fitted well for the majority of the subjects.

Time-series models such as ARX or Autoregressive Moving Average with Exogenous Input (ARMAX) require the model structure identification prior to the parameter estimation. The time series analysis for a human operator are utilized for example in [140, 116]. In [4] the Autoregressive Moving Average (ARMA) structure is used to model human operator dynamics during pursuit manual tracking. There were two different unpredictable input targets: sum of sine waves and random ternary signals. In all cases, ARMA models of order (2,1) proved to fit well the experimental data when the pure delay of the human operator response is left out. Different types of linear parametric models were tried to model the human driver model uncertainty from a driving simulator data in [29]. The simplest structures are ARX, but ARMAX structure gives residuals closer to white noise. The Box-Jenkins structure also yielded good residual whiteness, but it has more parameters than the ARMAX model. In [108] Multi Input Single Output (MISO) ARX models are used for pilot model identification, where the ARX method is compared with the Fourier coefficient method. The ARX method was found to be more accurate and in addition, it provides continuous frequency frequence estimation, whereas the Fourier coefficient method gives nonparametric estimation of the frequency response only at certain frequencies. In [109] ARX method is used to identify pilot's behavior in a roll-lateral helicopter hoover task. In [7] linear tranfer function models are fitted for experimental data where the tasks are: a point-to-point task (unit step), tracking of a sine wave, and tracking of a pseudorandom signal. Different types of recursive and static identification methods are tested. The second order linear models fitted rather well for the experimental data but, on the other hand, the controlled process was very simple. If more complicated process were used, probably a higher order model would have provided better fit to the experimental data.

The Optimal Control Model (OCM) is an example of a structural human operator model [81]. It assumes that a well-trained and motivated human operator behaves similarly to an optimal

controller. The model also accounts for the neuromotor limitations of the human operator. In [32], the Modified Optimal Control Model (MOCM) is proposed. The MOCM is modified from OCM so that the final linear transfer function can be expressed in a zero-pole form. Because the MOCM leads to an overparametrized representation of the human operator's dynamics the Fixed-Order Modified Optimal Control Model (FOMOCM) is proposed in [35]. The FOMOCM is based on the optimal projection equations and it allows the user to choose the desired order of the human operator model. In [10], the sensitivity model is described, which gives a systematic state-space representation like the optimal control model, but has the simplicity of the crossover model formulation. The sensitivity model is based on loop-shaping synthesis methods.

A nonlinear multiparameter model for a human operator is proposed in [71]. According to the results, the nonlinear model performed better than the second order quasi-linear transfer function model. In [30], a nonlinear driver model is identified by using the Nonlinear Autoregressive Moving Average with Exogenous Input (NARMAX) structure, where a nonlinear function is cascaded with a linear ARMAX transfer function. However, the non-linear structure is not very effective in reducing the model uncertainty. An adaptive model for a human controlling an aircraft in a situation when there is a failure or a sudden change in the active plant dynamics is proposed in [122]. The adaptive model response and the human operator are very close to each other until three seconds after the failure. After that time the human operator reponse is delayed of the model response. An iterative manual control model is proposed in [15]. The iterative model can be used to describe human's ability to learn while doing a repetitive task.

The human operator as a controller has also been researched in the brain science. According to the most important results reported in [67] the human operator acts as both feedforward and feedback manner as shown in Figure 3.5. As the skill of the human operator increases, the operator is able to predict the behavior of the controlled system better. Thus, the strength of the feedforward path becomes stronger. A nonlinear model for a human pilot controlling an aircraft exploiting the model structure shown in Figure 3.5 is proposed in [68]. In the model, the human operator acts both, as feedforward and feedback controller. The model is based on neural networks and it provides very accurate results when the aircraft control is simulated by using the model. In conclusion, it is argued that the feedforward path of a skilled human operator is nonlinear. However, since neural networks are known to be extremely powerful regression models, more careful validation and explanation of the model structure should have been introduced in [68]. The model structure shown in Figure 3.5 is also exploited in [43]. The human controller is divided into feedforward and feedback modules. Firstly the total closed-loop transfer function of the system is formulated. Using the reference signal and the output measurements, frequency domain identification methods are applied to obtain the human operator model. Closed-loop identification method is also used in [50, 69], where subspace identification is applied to identify the human operator model in a Multi Input Multi Output (MIMO) task. The model was able to capture the reference tracking properties of a human operator.

In [33] it is claimed that black-box or knowledge-based models such as Neural Networks or Fuzzy Logic models perform well in human driver modeling. The advantages of those model structures are that without changing the input-output components, different dynamics can be hidden inside the models. Nevertheless, it is claimed that a Fuzzy Logic model consisting of if-then expressions containing all parameters of a human driver would need approximately 10000-50000

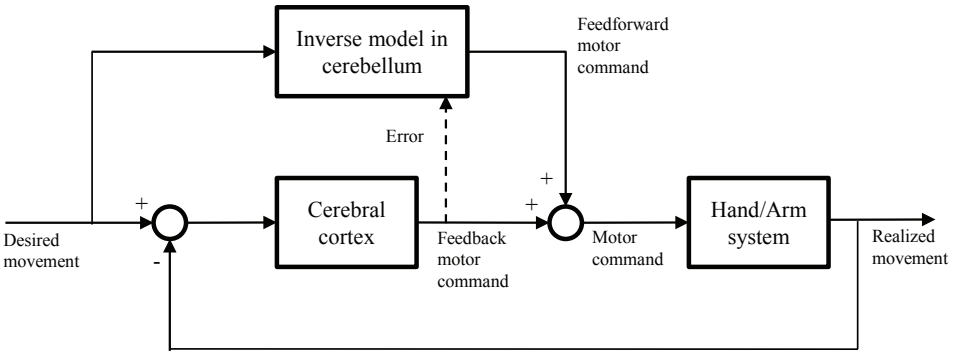


Figure 3.5: The structure of the Kawato-Suzuki-Ito model of the human controller. Adapted from [67].

rules to be identified. Therefore it is more useful to create models for some specific missions. In [37], an Adaptive Network-Based Neuro-Fuzzy Inference System (ANFIS) was compared to an ARX model for simple implementation to predict the human operator's response. The ANFIS model with only 32 rules provided better results than the ARX structure (with around 10 free parameters). However, in the ANFIS model with 32 rules there are significantly more than 10 parameters.

The human operation is typically intermittent. That is, the human does not continuously pay attention to the controlled system or control it continuously. Therefore, the analysis of the human operation from the perspective of intermittent control systems was performed in [27, 28, 70]. The results of modeling the human operator using the intermittent approach are promising but for practical use there is still research work to be done.

The structural methods for human operator modeling have been very effective to model a human operator in control tasks. In addition, one could argue that because they are based on the model of the controlled system, their parameters could be estimated based only on few trials of a task. In comparison to the models whose parameters are based purely on data, the training data should be descriptive enough in order to obtain reliable results. In terms of system identification, the training data should contain enough information for accurate high-order data-based models. On the other hand, the parameters of the linear parametric models are easy and efficient to identify.

From the point of view of working machines, the most appealing property for using the structural models is that they retain the mapping from the parameters of the controlled plant to the human operator model. Thus, they can be used to describe a continuum of models with respect to various operating points (e.g. different loads and rope lengths of a crane). However, currently the parameter identification of the MOCM-type models require the knowledge about the motor noise level, disturbance intensity of the human internal model, the singal-to-noise ratio of the closed-loop response of the human-machine system, the neuromotor time constant, and the effective time delay of the human operator. In addition, the measurements of the responses the human operator is assumed to use in control are needed to obtain the measurement

covariance matrices for the Kalman filter. In machine work, most of these these parameters cannot be assumed known and thus the identification leads to a direct-search-type iterative procedure. Therefore, this paper proposes two model structures, which are based on the optimal control models, but modified so that the parameters can be estimated from data by using gradient-based optimization methods. Thereafter, the models can be returned to the optimal control framework by solving the inverse optimal control problem. In addition, methods for dealing with the nonlinearities such as saturation and rate saturation in the identification phase are proposed. It is proposed that the nonlinearity of the human operator model can be based on the Hammerstein-Wiener structure. The human operator modeling is studied in Chapters 6 and 7.

3.3.4 Intelligent coaching systems

Because the coaching part is out of the scope of this thesis, here only a brief introduction to intelligent coaching systems is presented. Intelligent Coaching Systems (ICSs) and Intelligent Tutoring Systems (ITSs) have been researched widely in well-defined domains such as teaching mathematics or sciences [25]. The intelligent training systems based on ITS and ICS framework have shown the ability to speed up the learning of a student in learning evironments which use automatic instructors. The training system recognizes the individual training needs and the most important improvement areas of the human operator. Usually the training systems are based on virtual reality, where different scenarios can be easily implemented and the student's behavior and learning in those scenarios can be evaluated. In addition, the student can more easily see the effect of the actions and decisions in different situations.

However, when introducing an ICS to a real-world problem into the real application environment, for example, operating a working machine, the coaching problem often becomes ill-defined [6]. There may be a number of ways leading to equally or almost equally good solutions to the problem at hand. In addition, the intent of the operator needs to be estimated based on his/her actions and the measurements available.

A framework for using the intelligent coaching for mobile working machine operator is proposed in [119]. The structure of the system is shown in Figure 3.6. The basic structure of the system is similar to the most ICS described in the literature. In general, all ICSs have a performance interpreter, which evaluates the student's performance based on the available data. In addition, the systems have a learner model, which evaluate the learning of the student. Then there is a reasoning system, which decides the instructions for the student in order to increase the performance.

There are only a few succesful examples of implementing ICSs to the real application environment to continuously coach the human operator for better performance. Recently, in the car industry, two systems which can be understood as ICSs have been brought to market. The systems aim at increasing the fuel economy of the car by giving instructions based on the driver's driving style towards economic driving. However, the objective of the car driving can be stated clearly: safe and economic ride. Those two objectives are not conflicting. There are also only a few driving styles which lead to the best fuel economy. Thus, the fuel economy coaching problem is well-defined. In machine work, the domain is significantly more complex, and there are more objectives: productivity, quality, fuel economy, etc. Different operators

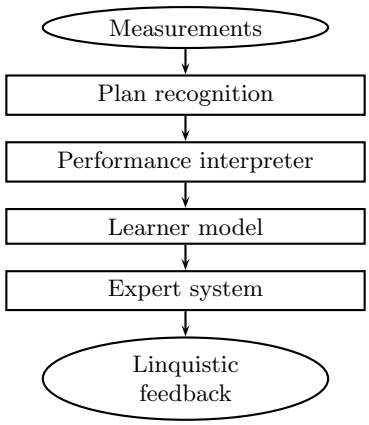


Figure 3.6: A flowchart of the intelligent coaching system for mobile working machine operators [119].

migh weight for example productivity and quality differently but still, when computing the earnings minus costs, the operators might obtain equal performance. In addition, they might have totally different working styles. Thus, the optimization problem becomes ill-defined. The author of this thesis has been involved in developing an ICS for forest machine operators, but it is still under development and thus left for future research. However, the work cycle modeling and the skill evaluation methods described in this thesis provide a solid basis for the ICS development. Besides, the HMI parameter optimization results shown in Chapter 8 can also be used as coaching instructions for the operator, if the parameters are not adapted automatically.

3.3.5 Online assistance and cooperative control

In the proposed HAMC the "human adaptation" can be either passive or active. Active adaptation means that the system intervenes the human operator's control signal either by modifying the dynamics of the control signal, or introducing a parallel human assisting feedback control, which accounts for the human operator's limitations. In the following, the human adaptive systems related to the framework of HAMC are shortly described.

Probably the first area where the online assistance for human controlled systems was considered was aerospace engineering. The optimal flight control synthesis described in [134] was called Cooperative Control Synthesis. It is a methodology which can be used to design control laws for piloted systems, based on available measurements for feedback control.

The structure of the cooperative control synthesis system is shown in Figure 3.7. The cooperative control is based on parallel structure, where the human operator controls the plant by using the information $\mathbf{y}$ obtained from the displays. The human control signal is denoted by δ. In addition, there is an automatic controller, which controls the system by using the measurements $\mathbf{z}$ available for feedback. The control signal of the automatic controller is denoted by δ_a. The final control signal u introduced into the plant is the sum of the human control signal

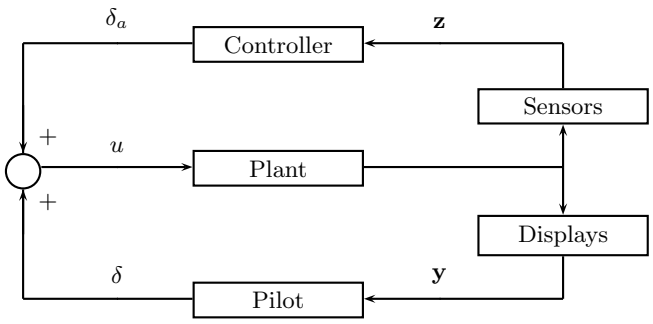


Figure 3.7: The structure of the cooperative control synthesis system. (Adapted from [31])

and the automatic control signal. The automatic controller is designed by using the OCM, or in extended cooperative control synthesis [31], the MOCM method.

The assistance control methods with similar idea as in the cooperative control have been proposed in [146, 145, 42]. The assist control force is calculated based on the human control model. The experimental device is controlled using the virtual internal model approach [84], and the parameters of the virtual internal model can be adapted based on the human operator model and skill level. In addition, the skill level is used to control the assist ratio, that is, the level the assistance is given to the operator. High skilled operators need less assistance.

An early survey for adaptive force assistant joystick controllers is provided in [130]. However, the methods presented there were based mostly on observing the characteristics of input and output trajectories and energies rather than modeling the human operator and designing the force reflection based the model. Similarly, in the areas of telemanipulation [104, 113] and intelligent wheelchairs [75, 103] different types of force assistant methods have been designed succesfully, but they do not consider the human operator's dynamic characteristics when designing the assistance.

The force assist control based on the estimation of human operator's stiffness is proposed in [102]. The human stiffness was estimated recursively based on the human interaction force and the differential of the human arm movements. The method increased the maneuverability of the teleoperation system significantly.

Various forms of operational assistance for operation of a rough terrain crane are proposed in [14, 169, 170, 171]. The assisting system is designed based on the user's emotional state (fatigue etc.) inferred using bionic signals, the skill level inferred from the oscillation and control history using recursive fuzzy inference, and the state of the crane and the payload. The operational assistance is given by showing multi display information using shadows, arrows and bars to help the user to understand the state of the system and the best control to supress the swinging. Moreover, haptic feedback was provided for the operator to communicate the best control action for the operator.

In the area of Human-Machine Collaborative Systems (HMCS) and teleoperation various types of assistance methods have been proposed to increase the maneuverability and task performance

of the human controlled systems. Virtual fixtures based on the intent recognition of the human operator increase the task performance of human controlled robots by removing the uncertainty and tremor inherent in human operations [3, 21].

Since the theretical development of the assist control is rather well covered especially in the earlier HAM research, and since when writing this thesis the actual implementation of the assist control in machine work environment was still under research, the active assistance interaction level is omitted in this thesis.

3.3.6 Passive and indirect human adaptation

Another "human adaptation" interaction level in the framework of the HAMC is the passive[8] adaptation. In passive adaptation the machine adapts itself to the behavior and control skills of the human operator. This is important because in many real-life systems the active adaptation is difficult to be realized, for many reasons. The measurements might not be good enough to obtain online intent recognition which is needed in the active adaptation. However, in passive adaptation it is enough that the intent is recognized after the task has been executed.

In practice, the passive adaptation means the adaptation of the static parameters such as the sensitivity and constraints of the HMI. There are only few examples of implemented passive adaptation systems in the literature, which can be interpreted as relevant for the scope of this thesis. For teleoperation, robust control tuning based on the human frequency response estimation is proposed in [128]. The human frequency response is estimated based on several executions of a given task. Then robust control tuning techniques are applied for designing the control law for teleoperation. Also the adaptation of the control is discussed, but no results are presented on that. Adaptation of the strength of the assistance given by an assistance system based on the human skill level is descibed in [170, 171, 146]. In all cases, the assistance is based on haptic feedback, whose strength could be controlled by the skill level of the human operator. Moreover, in [14], the time constant of the HMI is changed based on the user's skill obtained from the recursive fuzzy inference system and the emotional state estimation obtained using bionic signals.

In [62, 63], a subliminal calibration method is proposed to calibrate the dynamics of the machine towards the internal model of the human operator. The internal model here refers to the concept of human operator's comprehension of the dynamics of the controlled machine. It is proposed that the calibration is realized by identifying the human operator's internal model. The dynamics of the controlled system is then modified towards the internal model. They also use the concept of Just Noticiable Difference (JND) to ensure that the modification of the system dynamics does not disturb the human operator. That is, the calibration is done so carefully that the human operator does not perceive the change in the dynamics. Hence the name subliminal calibration. According to the results, the line tracking accuracy of the subjects improved when the calibration was used. However, the human internal model identification is done using neural networks, but the controlled machine is linear. There are risks when trying to modify parameteters of a linear system to behave as a nonlinear model (neural network-based human internal model in this case). Moreover, in practice modifying the machine dynamics is difficult, because the machine is a physical system. One should introduce a virtual model, which

[8] In this context, one should not confuse passivity with the control theoretic concept of passivity.

modifies the dynamics of true system plus the virtual model closer to the human operator's internal model. This was not discussed thoroughly in [62, 63]. However, the virtual internal model approach was decribed in [146], where the machine "XY-stage" was controlled using the virtual interal model method proposed in [84]. Then the parameters of the internal model were tuned according to the human model using a Lyapunov function-based method.

When only the static parameters of the HMI are tuned, it is easy for the operator to interpret and understand their implications. This is an important point from the usability perspective. If the human operator does not understand the effects of the adapted parameters, the *transparency principle* of good usability is violated [54].

In this thesis, the passive adaptation to the human operator's skill level and dynamic characteristics is realized by modifying the static parameters of HMI. The calibration, or passive adaptation, is approached from the controller design perspective. It is shown that the HMI design (at the simplest) can be converted into a very simple controller tuning problem: "given the human controller and the machine, find the interface parameters which fulfill the design criteria". In particular, the simplest case, that is, the sensitivity adaptation of the HMI, one does not need to change the control structure of the machine to the virtual internal model method, but the adaptation can be based purely on the current HMI. If the dynamics of the human control signal are desired to modify, the virtual internal model method is needed as proposed in [146]. When the human operator model is not available, one can use knowledge-based approach to adjust the HMI parameters. The knowledge-based approach exploits the skill evaluation method proposed in Chapter 5.

Moreover, in [63] it is argued that the internal model quantification cannot be done during the normal task execution, but it should be done during separate calibration test. In this thesis, a new method for implementation of the HMI parameter adaptation based on normal task execution data is proposed. The proposed Skill Adaptive Control (SAC) algorithm can be implemented for realizing the passive adaptation property of crane systems based on normal work data. The HMI adaptation is discussed in detail in Chapter 8.

4 Hidden Markov model-based work cycle modeling method

This chapter focuses on the intent recognition part of the HAMC, which is a prerequisite for the skill evaluation concept presented in Chapter 5. The intent recognition, as explained in Section 3.3.1, is a very general concept and the human intent includes significantly more than just the objective of the work. However, from the applications' perspective and the development of HAMC it is essential to recognize "what the human operator is trying to do and how well does he/she perform". In this thesis, the "what" part equals to the intent recognition and it is performed using HMM-based work cycle modeling method. Here, the intent represents the objective of the work and the associated task sequence performed to accomplish the objective.

As it was discussed in Section 3.3.1, the intent recognition can be performed using several methods such as DBN [150, 151, 149, 86], HDBN [135], LHMM [1, 2, 36], HMM [87, 22, 152, 174, 118, 155, 156], FHMM [88], HHMM [58], neural networks [46, 38], and fuzzy reasoning [64, 65, 77]. Moreover, the intent recognition can be performed using several different approaches and modification of the same method. For neural networks, fuzzy reasoning, and HMMs there exists readily available and widely used official toolboxes for training, analysis and use of the methods. There exists a good toolbox for Bayesian networks as well. Since the objective in the work cycle recognition is to classify dynamic time-series data into sequences, the dynamic models such as DBNs and HMMs are natural choices. In fact, HMM is only a special case of DBN. An HMM with one dimensional discrete observation probability distributions were chosen in this case because it provides very parsimonious but yet powerful classification of temporal context dependent signals and they are directly applicable for observation sequences of different lengths. In addition, using a single HMM with states corresponding the work tasks the computational cost of the task sequence estimation is very low.

This chapter proposes a general framework for work cycle modeling of human operated machines using HMMs. This chapter is mainly based on the author's Master's thesis [155] and the publications [156, 17, 18].

4.1 Hidden Markov Models (HMMs)

An HMM consists of two simultaneous stochastic processes. The first, underlying stochastic process constitutes a Markov chain, but unlike with ordinary Markov chain models, the states cannot be observed. The second stochastic process produces a sequence of observations. Each state has a probability distribution for the observations to appear. Thus, based on the sequence of observations the most probable respective state sequence can be deduced [125]. Due to the stochastic and dualistic nature of HMMs, they are very often used in modeling the human performance.

An HMM can be defined by the following characteristics [125]:

i. A finite number of unobservable (i.e. hidden) states. Set of possible states is denoted by $\mathcal{S} = \{S_1, S_2, \ldots, S_{N_s}\}$, where N_s is the number of states. Let the state at time t (discrete time index here) be denoted by $q_t \in \mathcal{S}$. Moreover, the notation $Q = q_1, q_2, \ldots, q_T$ is used to denote the state sequence within time interval $1 \leq t \leq T$.

ii. The state transition probability matrix $\mathbf{A} = \{a_{ij}\}$. Element $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ describes the probability of moving from state S_i to state S_j.

iii. An observation set $\mathcal{O} = \{v_1, v_2, \ldots, v_M\}$, where M is the number of possible observation characters (discrete-valued). The observation (emission) at time t be denoted by o_t. The observation sequence within the time interval $1 \leq t \leq T$ is denoted by $O = o_1, o_2, \ldots, o_T$.

iv. The emission probability matrix $\mathbf{B} = \{b_i(k)\}$, where $b_i(k) = P(o_t = v_k | q_t = S_i)$. Each row of $\mathbf{B}$ forms an observation probability distribution of that particular state.

v. The initial state distribution $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = P(q_1 = S_i)$, i.e. the probability that the system starts from state S_i.

The compact notation $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ is often used to represent an HMM. In general, the observation sequences can be more than one dimensional, and the observation probability distributions can be continuous. In this thesis however, the discussion is restricted to HMMs with discrete observation probability distributions and one dimensional observation vectors.

With HMMs, there are three basic problems of interest [125]: First, the training problem, training the parameters of the model $\boldsymbol{\lambda}$. Secondly, the evaluation problem, evaluating the probability of the observation sequence O being produced by the model $\boldsymbol{\lambda}$. Thirdly, the decoding problem, decoding the most likely path of states q that maximize the probability of the observation sequence O. In the following, the corresponding problems of interests are desribed in terms of the work cycle recognition.

4.2 Work cycle modeling by HMMs

4.2.1 Definitions, assumptions and requirements

The objective in the work cycle modeling is to infer the (unobservable) intent of the human operator based on the (observable) measurements. In machine work, the intent represents the task, task sequence, or objective the human operator desires to accomplish. To be exact, the intent and the tasks are not interchangeable terms: one could say that the tasks are instances of the intent. However, in this thesis the two terms are used interchangeably.

The basic idea behind the work cycle recognition using HMMs is illustrated in Figure 4.1. On the lowest level, the measurements are obtained from the machine work. The measurements can be multidimensional, continuous, discrete, etc. depending on the application. By using the measurement data, one can extract features, which are designed to separate the tasks in the work cycle recognition. In this thesis, it is assumed that the features are basically integer-type events, whose values refer to different states of the process.

The tasks represent the highest level in the work cycle modeling. Each task is designed to correspond to a state of the HMM. Thus, each task has a probability distribution $\mathbf{B}$ for different observations to appear in that particular state. The typical flow of tasks is encoded into the state transition probability distribution matrix $\mathbf{A}$. One single event o_t does not, in general, contain enough information for the task classification, because the same event can possibly appear in several states. However, based on the sequence of events O, the most probable

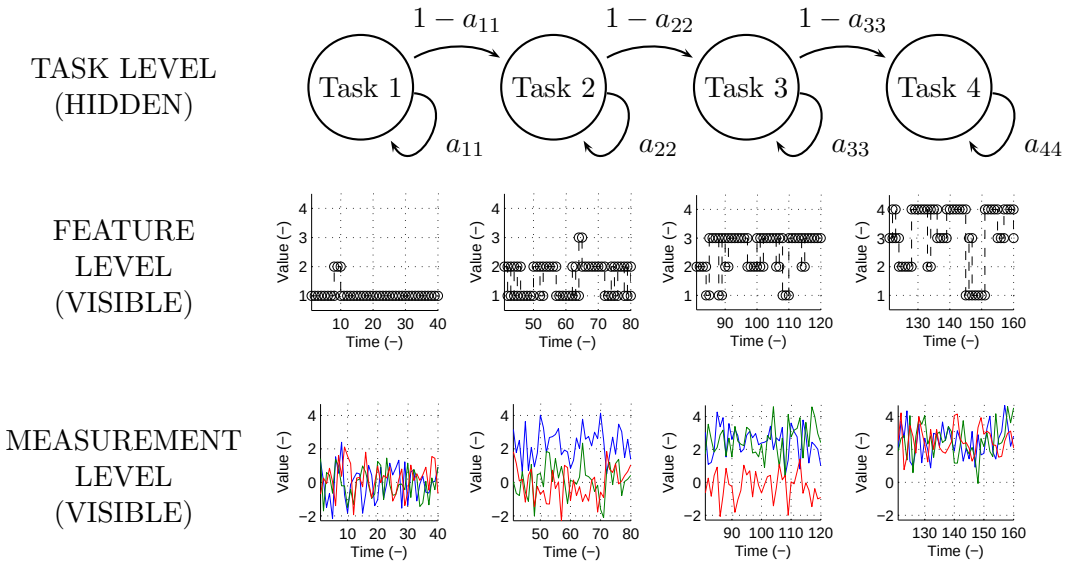


Figure 4.1: Illustration of the levels in the HMM-based work cycle modeling method.

corresponding state sequence can be calculated given the parameters of the HMM by using the Viterbi algorithm.

The work cycle of the human operated machine can be modeled by using the HMM-based approach if the work of the machine fulfills the following assumptions.

1. Repetitivity of the work: The work the human operator performs with the machine must be repetitive. That is, successive work cycles should have common features and tasks.

2. Small number of tasks: The work cycle of the machine should consist of relatively small number of tasks.

The first assumption, repetitivity, is required because the state transition probability distribution of the HMM is designed to capture the task hierarchy in the work cycle modeling method. The HMM should be of left-to-right type, meaning that the tasks constitute a hierarchy. In normal work, the task hierarchy should not be broken. If the work is not repetitive, the tasks do not form any hierarchy, and thus there is no advantage of using the dynamic HMM-based classification.

The small number of tasks assumption is due to the classifcation capability. This assumption is relative to the available information in the measurement set the task sequence classification is based on. If some tasks are related to each other in terms of the signal values and dynamics used for classification of the tasks, the work cycle model might not be able to separate these tasks if they are not significantly different in terms of the task hierarchy.

4.2.2 Measurements and feature extraction

As in any classification and modeling methods, also in the work cycle modeling there is no general solution for the feature extraction problem. In all practical cases the method was applied during this thesis, the features were discrete integer-valued events. Different values of the events represent the different states of the system. The features are extracted based on the multidimensional measurement data available in the machine's database or in the distributed control system. Discrete one-dimensional features were used because they resulted in equal or better classification performance than continuous-valued multi-dimensional features. In addition, the training and decoding algorithms are significantly more efficient for one-dimensional than for multi-dimensional cases.

Examples of the possible features are shown in Figure 4.1, whose data are generated by using the code in Appendix A.1. On the measurement level, there are three continuous signals. The event values (from 1 to 4) describe the different combinations of high values for the three sigals. The simulation example is discussed in detail later in this chapter.

4.2.3 Estimation problem: training

The parameter estimation problem of the work cycle modeling can be performed by using supervised or unsupervised learning, depending on the available measurements. For supervised learning, one should have readily labeled training data. The training data can be labeled for example by using video recordings from the work, or some other additional measurements. In any case, for the supervised learning one needs to label the data manually. Once the data are labeled, the parameters of the HMM can be obtained by using simple relative frequency-based probability estimators as described in [155].

In the unsupervised learning the training data need not to be labeled, but one needs good initial guesses for the HMM parameters $\boldsymbol{\lambda}$. It is very important that the state transition probabilities are set according to the desired hierarchy of the tasks. In addition, the probabilities of the forbidden transitions should be set to zero. The unsupervised estimation can be carried out by using the Baum-Welch method [125] or the Viterbi re-estimation [161] method. In practice, the Viterbi re-estimation turned out to be better for the work cycle model training, because it results in parameters which give a smoother state sequence, and it is significantly faster than the Baum-Welch method.

The supervised training method results in state definitions, which correspond to the tasks of the work. On the contrary, the unsupervised training can converge to the parameters, which do not result in desired state definitions. Therefore, the importance of choosing the correct initial values needs to be emphasized. In addition, the work cycle model needs to be validated. The validation of the model is discussed later in this chapter.

4.2.4 Decoding problem: task sequence estimation

In terms of the work cycle recognition, the decoding problem corresponds to finding the most likely task sequence given the observations O and the model $\boldsymbol{\lambda}$. An efficient solution for the decoding problem is the Viterbi-algorithm [164]. The Viterbi-algorithm formulation for HMM

can be found from [125]. The algorithm is readily implemented for example, in Matlab Statistics Toolbox command "hmmviterbi".

4.2.5 Evaluation problem: task type and anomalous work cycles

The evaluation problem of HMMs is to find the probability of the observation sequence O given the HMM $\boldsymbol{\lambda}$. In practice, it can be done efficiently using the forward-backward algorithm [125]. In terms of the work cycle modeling, the evaluation problem has several uses.

If the basic task types (loading, unloading, etc.) cannot be detected deterministically from the measurement data in normal work, the evaluation problem can be used to solve the task type recognition problem. If separate models are trained for each task type, say, $\boldsymbol{\lambda}_{\text{loading}}$ and $\boldsymbol{\lambda}_{\text{unloading}}$, one can recognize the task type of the observation sequence O by evaluating

$$\arg\max_i P(O|\boldsymbol{\lambda}_i). \tag{4.1}$$

Another problem which arises in the work cycle modeling is the recognition of anomalous work cycles. That is, the main assumption of the HMM-based method is that the successive work cycles are similar at least in the sense of the observation sequences. However, because the machine is controlled by the human operators, several unexpected situations may appear. Occasionally, the human operator can perform some rare task types, which cannot be modeled reliably by using the trained HMM. In addition, those work cycles can be so rare that developing an own model for them would not be rational. The most useful solution is to recognize the anomalous work cycles and classify them as "unclassified work".

Basically, the anomalous work cycles can be recognized by defining a threshold for the normalized log-likelihood

$$\text{LogP} = \frac{1}{T}\log\left(P(O|\boldsymbol{\lambda})\right), \tag{4.2}$$

where T is the number of observations in the work cycle O. The normalization is required because the observation sequences can have different lengths. In practice, however, LogP did not turn out to be a reliable measure for classification of anomalous work.

Thus, a measure based on the state transition probabilities was developed. The StateP metric is defined as follows

$$\text{StateP} = \sum_{t=1}^{T-1} \log(a_{q_t,q_{t+1}} \mid q_t \neq q_{t+1}). \tag{4.3}$$

The StateP sums the logarithmic state transition probabilities where the following state at $t+1$ differs from the preceding state at t. Because the work cycle modeling method assumes that the normal work of the machine follows the task hierarchy encoded into the state transition probability matrix $\mathbf{A}$ (e.g. state transition diagram in Figure 4.1), there should be only state transitions with high probability in the most likely path of states given by the Viterbi algorithm. If the state path Q contains abnormal state transitions, StateP measure reacts to those efficiently.

In practice, a threshold value is needed for the classification of normal and abnormal StateP values. For example, one can compute the StateP values for several thousands of work cycles

and then choose the threshold so that most of the abonormal work cycles could be separated from normal ones. In addition, one can check by hand the state transition sequences of the work cycles close to the threshold and then choose the threshold so that the misclassification probability minimizes.

4.2.6 Model validation

As in any modeling and classification, also in the work cycle modeling the validation is an important part of the model building process. During this work, a visual video-based validation scheme was developed together with Lauri Palmroth, which was first implemented for the forest harvester and then for different types of cranes. Firstly, the data are synchronized with the video material. Then, the work cycle modeling results (task names) are imprinted onto the video as subtitles. Thus, a convenient method for validation of the recognition performance is obtained.

In addition to the visual validation, a quantitative validation can be obtained by using the confusion matrices. The confusion matrix gives estimates of the misclassification probabilities of the HMM. Let the true state sequence (labeled) of a work cycle be denoted by $Q = q_1, q_2, \ldots, q_T$. In addition, let the estimated state sequence be denoted by $\hat{Q} = \hat{q}_1, \hat{q}_2, \ldots, \hat{q}_T$. The estimate for the classification accuracy can be obtained by the confusion matrix

$$\mathrm{CM}_{i,j}^{\mathrm{events}} = 100\% \times \frac{\sum_{t=1}^{T} \delta_{q_t,i}\delta_{\hat{q}_t,j}}{\sum_{t=1}^{T} \delta_{q_t,i}}, \tag{4.4}$$

which is an identity matrix, if the classification is perfect[9]. However, in the work cycle modeling the misclassification of mere events is not that interesting. As the objective is to classify the work of the machine into tasks, a more interesting quantity is the amout of time classified correctly or incorrectly. For this purpose, let $\mathrm{TIME}(t)$ denote the real-world time at the observation time instant t. Thus, the confusion matrix with respect to time becomes

$$\mathrm{CM}_{i,j}^{\mathrm{time}} = 100\% \times \frac{\sum_{t=2}^{T} \left(\mathrm{TIME}(t) - \mathrm{TIME}(t-1)\right) \delta_{q_t,i}\delta_{\hat{q}_t,j}}{\sum_{t=2}^{T} \left(\mathrm{TIME}(t) - \mathrm{TIME}(t-1)\right) \delta_{q_t,i}}, \tag{4.5}$$

which is also an identity matrix in case of perfect classification. In both cases, the element (i, j) describes the case the true state is S_i and the estimated state is S_j.

4.2.7 General procedure for constructing the work cycle model

Based on the experience from several case studies, a general procedure for work cycle modeling using HMMs can be formulated as follows.

1. *Break the work to tasks*: divide the work to tasks based on the knowledge of the domain experts. Identify the most important main task types (loading, unloading, etc.), the most important tasks, task sequence and the task hierarchy.

[9] Two argument Kronecked delta is $\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$.

2. Study the available measurement signals for modeling: control lever motions, button presses, positions, speeds, angles, etc.

3. *Observation design*: based on the task hierarchy and the signals, design a set of events, which can separate the tasks as well as possible. It is important that the tasks next to each other in the task hierarchy can be separated by at least one event.

4. *Data gathering*: gather data for model training and validation. Preferably one should have video matrial from the same work period. Best results are achieved when the data are gathered during normal work of professional operators.

5. *Model training and validation*: train the model using either supervised or unsupervised method. Validate the model using the confusion matrix method and the video analysis.

6. *Implementation*: once the model works reliably, it can be implemented in the machine's operating environment.

In the first step, the feasibility of the method for the particular application should be decided. If the work is repetitive and consists of distinct tasks which can be separated, then the work is suitable for the HMM-based approach.

4.3 Simulation study

4.3.1 Work cycle model training

For illustration of the work cycle modeling method, a simulation is developed. In the simulation, three continuous singals are used as inputs for work cycle modeling. The signals could, for example, represent the human control signals for a three degrees-of-freedom crane. The signals are normally distributed standard random signals whose levels are changed in different states (tasks). The three signals constitute the measurement level of Figure 4.1. By monitoring the levels of the signals, four observations are generated. The observations constitute the feature level of Figure 4.1. The states are recognized based on the combination of levels of different signals. The task hierarchy of the simulation model is shown in Figure 4.1. The tasks are defined to correspond to the states of the HMM. There are four tasks, which follow the left-to-right hierarchy. The tasks constitute the task level of Figure 4.1.

The data generation function code is presented in Appendix A.1. In addition to the illustration purpose, this example can serve as a benchmark for alternative work cycle modeling solutions which might appear in the future.

For features, four events based on a single threshold for each signals are designed as follows.

1. The values of all signals are low.

2. The value of signal 1 is high and the values of signals 2 and 3 are low.

3. The values of signals 1 and 2 are high and the value of signal 3 is low.

4. The values of all signals are high.

Ten realizations of the random process were generated and the corresponding events are extracted. By using the generated data, two HMMs are trained. One by using the supervised training and the other by using the (unsupervised) Viterbi re-estimation procedure. In the latter case the initial guesses for the HMM matrices are as follows

$$\mathbf{A}_{\mathrm{g}} = \begin{bmatrix} 0.990 & 0.010 & 0 & 0 \\ 0 & 0.990 & 0.010 & 0 \\ 0 & 0 & 0.990 & 0.010 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{B}_{\mathrm{g}} = \begin{bmatrix} 0.800 & 0.200 & 0 & 0 \\ 0.200 & 0.700 & 0.100 & 0 \\ 0.100 & 0.200 & 0.600 & 0.100 \\ 0.100 & 0.100 & 0.200 & 0.600 \end{bmatrix},$$

where the subsript "g" stands for "guess". The HMM matrices after the training for supervised and unsupervised cases are as follows

$$\mathbf{A}_{\mathrm{sup}} = \begin{bmatrix} 0.975 & 0.025 & 0.000 & 0.000 \\ 0.000 & 0.975 & 0.025 & 0.000 \\ 0.000 & 0.000 & 0.975 & 0.026 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \qquad \mathbf{B}_{\mathrm{sup}} = \begin{bmatrix} 0.974 & 0.020 & 0.005 & 0.000 \\ 0.336 & 0.655 & 0.018 & 0.000 \\ 0.126 & 0.328 & 0.532 & 0.018 \\ 0.064 & 0.135 & 0.253 & 0.550 \end{bmatrix}$$

$$\mathbf{A}_{\mathrm{uns}} = \begin{bmatrix} 0.971 & 0.029 & 0.000 & 0.000 \\ 0.000 & 0.977 & 0.023 & 0.000 \\ 0.000 & 0.000 & 0.976 & 0.024 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \qquad \mathbf{B}_{\mathrm{uns}} = \begin{bmatrix} 0.977 & 0.023 & 0.000 & 0.000 \\ 0.354 & 0.625 & 0.021 & 0.000 \\ 0.099 & 0.332 & 0.549 & 0.020 \\ 0.070 & 0.102 & 0.217 & 0.611 \end{bmatrix},$$

where the subscripts "sup" and "uns" are for supervised and unsupervised cases, respectively.

4.3.2 Work cycle model validation

For validation of the trained models, ten more realizations of the signals are generated. The events are extracted from the signals and the resulting event sequences are used as inputs for the trained HMMs. The results given by the supervised and unsupervised models for a realization of the validation data can be seen in Figure 4.2. Comparison of the true state sequence and the estimated state sequences reveals that both models recognize the state changes some events late. The reason for this can be easily understood when looking at the state transition and observation probability matrices. For both models, the observation value 1 is more probable in state 1 (0.974 for supervised and 0.977 for unsupervised) than in state 2 (0.336 for supevised and 0.354 for unsupervised case). Thus, the models will not transfer to the second state until the first observation value 2 is obtained.

By using the ten generated validation data realizations, the confusion matrices were computed for both models using (4.4). The resulting matrices are shown in Table 4.1. According to the matrices, the supervised and unsupervised models have equal performance with respect to the first and the second states. Both models provide over 99 % classification performance. As for the third state, the supervised model results in the classification capability of 94.25 %, whereas the unsupervised model gives 95 %. The supervised model recognizes the fourth state slightly better than the unsupervised one. All in all, both models result in an equal performance in this case.

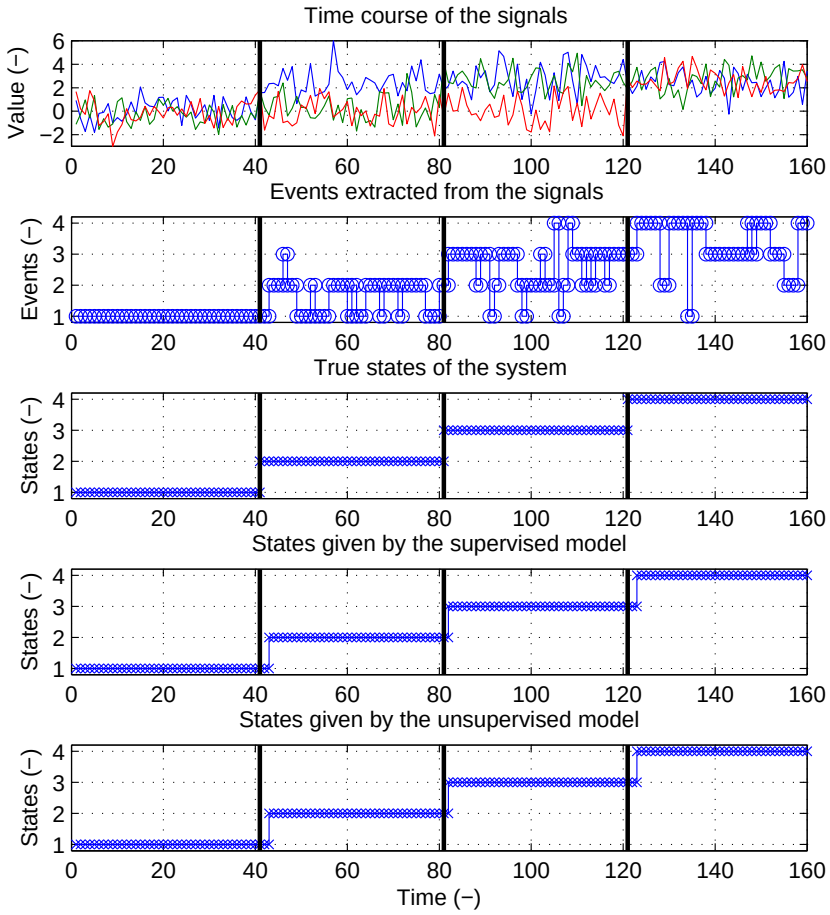


Figure 4.2: A realization of the simulation model signals, the corresponding events, states and the results given by the models. This realization was not used for training the models.

4.4 Case: STS work cycle modeling

4.4.1 Work cycle model training

In order to obtain an understanding of the different aspects in STS work, two visits to a harbor site were arranged and the operators of the crane were interviewed. During these visits, raw data were gathered from the database of an STS. In addition, the work of an STS was filmed on a video, which can be used for validation of the work cycle model. The description of the STS and the measurement capability are presented in Section 2.4.2. The basic work cycle of the cranes studied in this thesis is presented in Section 2.6.

During the model development process, several structures and task definitions were tried. The final work cycle model structure of the STS is shown in Figure 4.3. A new work cycle starts when the previous container is released. Then the empty spreader is moved towards the container.

Table 4.1: The confusion matrices of the simulation example based on state transition frequencies (4.4). On the left, the supervised, and on the right, the unsupervised case.

	$\hat{S}_1^{\text{sup}}$	$\hat{S}_2^{\text{sup}}$	$\hat{S}_3^{\text{sup}}$	$\hat{S}_4^{\text{sup}}$
S_1	99.00	1.00	0.00	0.00
S_2	0.75	99.25	0.00	0.00
S_3	0.00	3.50	94.25	2.25
S_4	0.00	0.00	9.25	90.75

	$\hat{S}_1^{\text{uns}}$	$\hat{S}_2^{\text{uns}}$	$\hat{S}_3^{\text{uns}}$	$\hat{S}_4^{\text{uns}}$
S_1	99.00	1.00	0.00	0.00
S_2	0.75	99.25	0.00	0.00
S_3	0.00	3.50	95.00	1.50
S_4	0.00	0.00	9.75	90.25

There migh be some waiting during the movement. The container approaching is defined as a separate task from the moving task, because it is typically performed more carefully than pure moving. The left-to-right model structure is broken because of the waiting states, which can be understood as "parallel" to the active moving states in the task hierarchy. The recognition of the waiting states is, in practice, deterministic. Observation of a stopping event practically always causes the model to transfer to a waiting state. The task hierarchy does not allow backwards transitions with respect to the moving states directly. Once the container is caught, the task sequence proceeds from lifting, moving, approaching landing, to the placing container state, where the container is released. At this point, the work cycle ends, and the next work cycle starts. Again, during the moving states, there might be some waiting, which are recognized as separate tasks.

For constructing the STS model, raw data from two cranes from different harbors were available. From now on, the cranes are referred as STS1.1 and STS2.1. One month data from both harbors were selected as the training data. In total, roughly 4000 cycles. As for the features, 13 distinct observations were generated based on different combinations of the speed references and the position values of all three degrees of freedom of the movements, as well as the binary information regarding whether the crane is holding a container, or whether the corner locks of the spreader are on or off.

The model training was carried out by using the Viterbi re-estimation procedure with the initial guesses for the HMM matrices chosen based on the desired task hierarchy and task definitions, similarly as in the previous simulation study. In the Viterbi re-estimation procedure it is very imporant to provide reasonable initial guesses for the matrices $\mathbf{A}$ and $\mathbf{B}$ so that the algorithm would converge to parameters which provide a desirable state hierarchy. Typically in the work cycle models the probabilities of self-transitions a_{ii} are high and the probabilities of the forbidden states are zero(in the training phase). In the task hierarchy shown in Figure 4.3 the work cycle always terminates to the 12^{th} state, and therefore the corresponding state transition probability was set to $a_{1212} = 1$. The other transition probabilities can be set based on the task hierarchy of the work cycle similarly as in the simulation example in Section 4.3.1. The observation probabilities in matrix $\mathbf{B}$ were set based on the definitions of the observations and the states. It is assumed that the work cycle always starts from the first state. Thus the only nonzero value in the initial state probability distribution is $\pi_1 = 1$. The practical implementation with respect to algorithms and computational conditions were similar as with the simulation study in Section 4.3.1 and the Viterbi training in Appendix A.2.

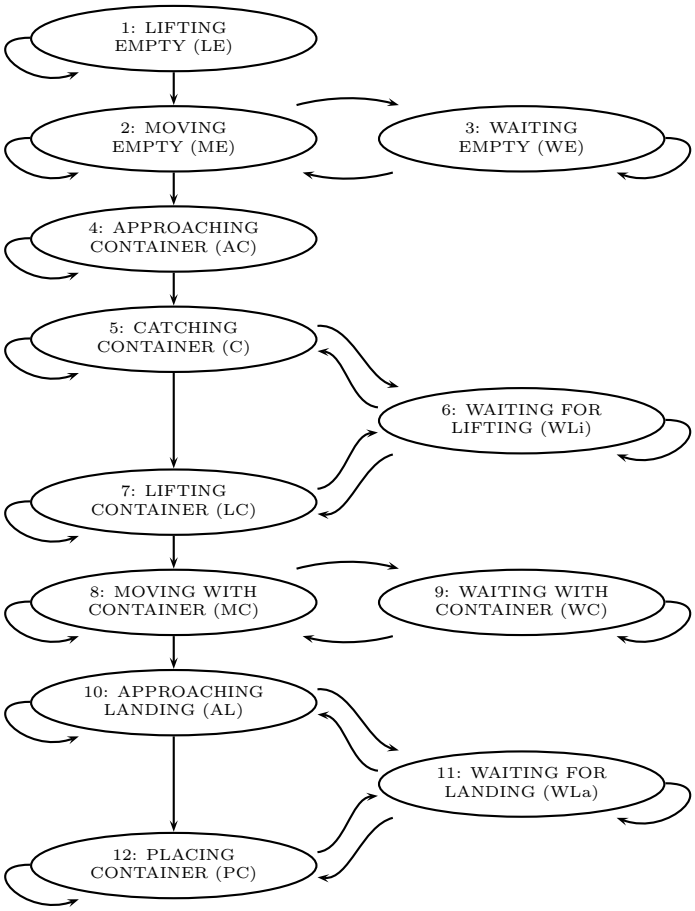


Figure 4.3: The task hierarchy, the most probable transitions in STS and RTG work cycle, and the associated task name abbreviations used in Table 4.2 and Table 4.4.

The HMM probability distributions given by the Viterbi training procedure are shown in Figure 4.4. One can note that the state transition probability matrix is mostly diagonal, except for the waiting states. In the present form, the waiting events are emitted only when the waiting begins. Thus, the model transfers to the waiting states once, but the successive observations is always some type of a moving event. Thus, the self transition probabilities for the waiting states are zero. One can also note that the first event, that is, the stopping event is, in general, accepted only in the waiting states. Note that in practice when using the model, one should set either the nonzero elements in **B** or **A** to a very small probability in order to prevent the numerical crash of the state path estimation due to an observation sequence which does not conform to the trained model.[10] Such observation sequences may occur due to work cycles which do not follow the task hierarchy of Figure 4.3, or due to measurement noise. The

[10]The numerical crash occurs in a case when the likelihoods of all paths tend to zero. In the implementation of the Viterbi algorith, the logarithm of the path likelihood is required, which is (negative) infinite at zero.

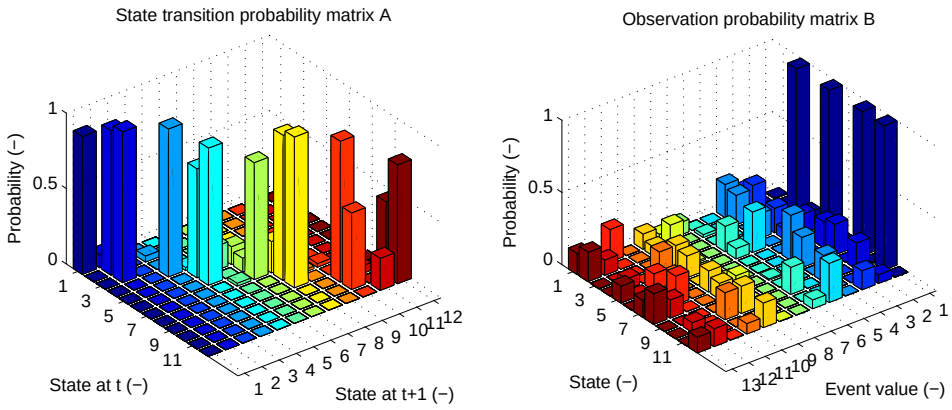


Figure 4.4: The state transition probability matrix **A** and the observation probability matrix **B** of the STS work cycle model.

recognition of anomalous work cycles in the STS case is discussed in Section 4.4.3.

4.4.2 Work cycle model validation

The initial validation for the work cycle model was performed by using video material, where the states given by the model were imprinted as subtitles. The video was recorded by using four cameras from different angles. The first one was in the STS cabin following the joystick commands of the operator, the second in the cabin following the spreader movements (the operator's view), the third on the quay following the crane movements and the last one on the quay following the straddle carrier traffic. The four angles provide a comprehensive overview of the different aspects in STS work during the work cycle. A screenshot of the video is shown in Figure 4.5. Based on the video presentation it was rather easy to discover the most important flaws of the model in the development phase.

In addition to the video-based validation, the confusion matrix method was used. Raw data from one STS were analyzed manually to define the developers' comprehension about the states of the crane. The state sequences obtained by the manual analysis were compared against the state sequences given by the work cycle model.

The STS has two different task types, loading and unloading. Five work cycles from both task types were randomly selected for the STS confusion matrix analysis. The confusion matrix of the STS computed using (4.5) is shown in Table 4.2. Only the time-based confusion matrix is computed, because in the harbor crane case, the interest is to analyze the bottlenecks of the harbor work. According to the results, the HMM-based work cycle model agrees well with the manually defined state sequences.

Here, it must be emphasized here that the number of analyzed work cycles was low and the manual data labeling was based on the "developers' comprehension" about the work cycle (only two people). Thus, in order to better validate the results more analysts should be used in the

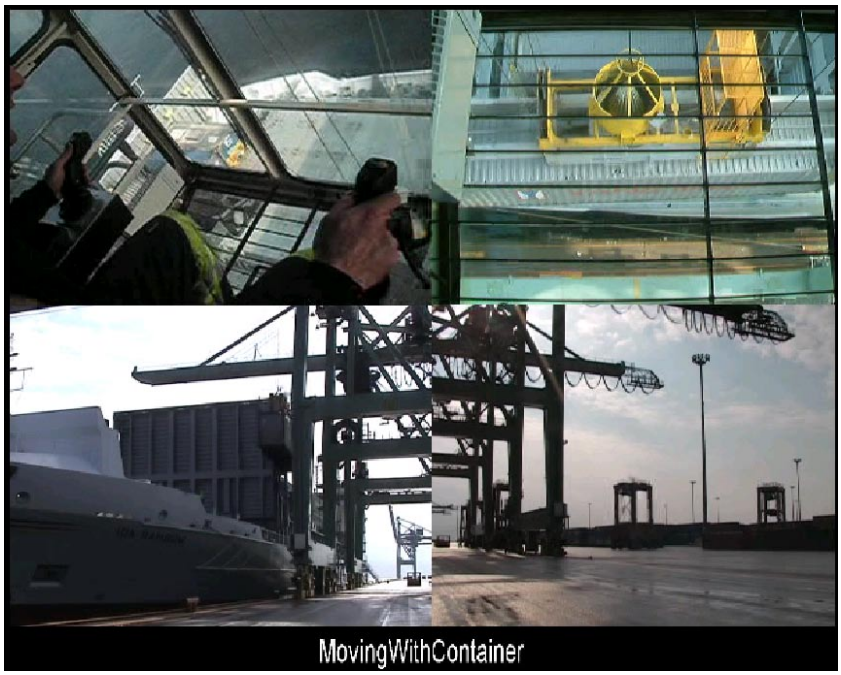


Figure 4.5: A screen shot from the video presentation. [17]

data labeling so that the systematic error due to individual analyst could be averaged away. On the other hand, the duration of a work cycle is typically more than two minutes. The short tasks such as catching and releasing can be seen pretty easily from the time-series of the measurement data (and the video synchronized with the data). Thus, labeling of those tasks should not cause significant analyst-dependent error to the results. The recognition of the moving and approaching tasks are more sensitive from that aspect. However, the durations of those tasks are typically more than 10 seconds. Thus differences in the magitude of fractions of seconds between the individual analysts does not have too high impact on the accuracy of the classification. Nevertheless, in order to get truly reliable validation results, several analysts and hundreds of work cycles should be used in the validation.

4.4.3 Recognition of anomalous work cycles

A threshold for the StateP was defined by performing the HMM-based for several thousands of cycles using the raw data from both harbors (STS1.1 and STS2.1). Based on the resulting state sequences, the StateP value was calculated for every work cycle using (4.3) and plotted against the container number as shown in Figure 4.6. One can see that there is a clear gap between about $(-5.80, -4.30)$, from which range the threshold for work cycle rejection crierion can be defined. The threshold was set to -4.50.

The work cycles obtaining a StateP value smaller than the threshold are classified as “abnormal work”, and they are omitted from further analyses of the result data. The numerical classifica-

Table 4.2: Time percentage confusion matrix of the STS based on the state time distribution (4.5). The abbreviations of the task names are given in Figure 4.3.

	LE	ME	WE	AC	C	WLi	LC	MC	WC	AL	WLa	PC
LE	86	14	0	0	0	0	0	0	0	0	0	0
ME	0	99	0	1	0	0	0	0	0	0	0	0
WE	0	0	100	0	0	0	0	0	0	0	0	0
AC	0	0	0	100	0	0	0	0	0	0	0	0
C	0	0	0	0	96	0	4	0	0	0	0	0
WLi	0	0	0	0	0	100	0	0	0	0	0	0
LC	0	0	0	0	0	3	87	10	0	0	0	0
MC	0	0	0	0	0	0	0	98	0	2	0	0
WC	0	0	0	0	0	0	0	0	100	0	0	0
AL	0	0	0	0	0	0	0	0	0	98	2	0
WLa	0	0	0	0	0	0	0	0	0	0	100	0
PC	0	0	0	0	0	0	0	0	0	0	6	94

Table 4.3: Work cycles of STSs classified as normal work based on the StateP measure.

STS	**Task type**	**Containers**	**Classified as normal work**	%
	Unloading	5222	5208	99.73
STS 1.1	Loading	5032	4989	99.15
	Unloading	2849	2778	97.51
STS 1.2	Loading	3217	3147	97.82

tion of the normal and abnormal work according to the StateP measure are shown in Table 4.3. Clearly, most of the cycles for STS1.1 can be categorized as normal work. For STS2.1 roughly 97.7 % of the cycles can be classified as normal work. The cycles classified as abnormal work can have anomalies due to two reasons: the work itself can be abnormal, or the measurements and events may obtain unexpected values in an unexpected order. An abnormal work cycle might, for example be a following loading cycle. The crane catches the container and transfers in to the seaside, but in the end, the container is transferred back to the shoreside and released there. This will produce a state sequence which contains a forbidden state transition, and thus the StateP value will be low. Such a cycle can occur when the land staff notices in the last minute that they are loading a wrong container.

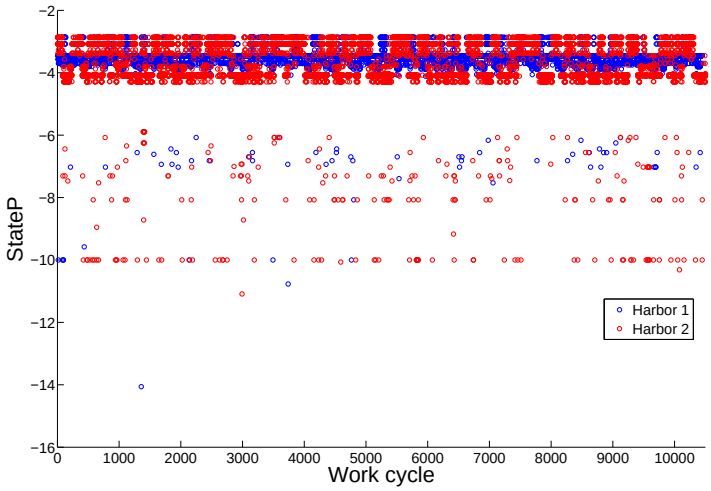


Figure 4.6: StateP values of the STS work cycles.

4.5 Case: RTG work cycle modeling

4.5.1 Work cycle model training

The description of the RTG is presented in Section 2.4.3 and the description of the crane work in general in Section 2.6. For the RTG case, raw data from four different RTGs from three different harbors were available for the estimation and validation of the model. In the following, harbors and RTGs are referred as RTGs 1.1 (harbor 1), RTG1.2 (harbor 1), RTG2.1 (harbor 2), and RTG3.1 (harbor 3). There were eight months of raw data available for RTG1.1 and RTG1.2, and one month data set for RTGs 2.1 and 3.1. For the training data set, in total 4204 work cycles from RTG1.1 were chosen.

The structure of the RTG model is same as the STS model shown in Figure 4.3. Only small modifications were made to the thresholds of some events, because the raw signals of the RTG have different scales. The initial guesses for the HMM matrices were carefully set to follow the task hierarchy shown in Figure 4.3. The training of the HMM was carried out by using the Viterbi re-estimation procedure similarly as with the STS case described in Section 4.4.1.

4.5.2 Work cycle model validation

For validation of the RTG model, no video material was available and at the time of writing of this thesis, no visits could be arranged to the harbor where RTGs are used. Thus, the validation of the model had to rely on the confusion matrix method where the state sequence was defined by manually analyzing the raw data of the crane. For the confusion matrix, raw data from one RTG were analyzed manually and the corresponding state sequences (according to the developers' comprehension of the states) were defined.

Table 4.4: Time percentage confusion matrix of the RTG based on the state time distribution (4.5). The abbreviations of the task names are given in Figure 4.3

	LE	ME	WE	AC	C	WLi	LC	MC	WC	AL	WLa	PC
LE	97	3	0	0	0	0	0	0	0	0	0	0
ME	0	99	1	0	0	0	0	0	0	0	0	0
WE	0	0	100	0	0	0	0	0	0	0	0	0
AC	0	0	0	100	0	0	0	0	0	0	0	0
C	0	0	0	14	86	0	0	0	0	0	0	0
WLi	0	0	0	10	0	89	0	0	1	0	0	0
LC	0	0	0	0	0	0	96	4	0	0	0	0
MC	0	0	0	0	0	0	0	100	0	0	0	0
WC	0	0	0	0	0	0	0	0	100	0	0	0
AL	0	0	0	0	0	0	0	0	0	96	0	4
WLa	0	0	0	0	0	0	0	0	0	0	100	0
PC	0	0	0	0	0	0	0	0	0	3	0	97

The RTG three basic task types are: truck to stack, stack to truck, and stack to stack. Five work cycles from each task type were randomly selected for the RTG confusion matrix. The confusion matrix computed using (4.5) is shown in Table 4.4. According to the results, the HMM-based work cycle model, in general, classifies the task times correctly. However, the same notions regarding to the reliability of the validation discussed in the STS case in Section 4.4.2 apply in the RTG case as well.

4.5.3 Recognition of anomalous work cycles

Similarly as in the STS case, a threshold for StateP was defined based on several thousands of work cycles. By using the threshold, the work cycles of each RTG were classified as normal, or abnormal work. The results are shown in Table 4.5. According to the results, the RTG work has very few anomalies. In the worst case, that is, RTG3.1 stack to stack roughly 98.5 % of the work cycles can be classified as normal.

One can interpret the StateP classification results so that the RTG work fulfills well the repetitivity assumption of the HMM-based work cycle modeling. However, one needs to be careful when interpreting the classification results. The quantities in Table 4.5 do not provide any insight of the classification capability of the model. They only give an estimate of the quantity of cycles which are normal work in terms of the task hierachy shown in Figure 4.3.

Table 4.5: Work cycles of RTGs classified as normal work based on the StateP measure.

RTG	Task type	Containers	Classified as normal work	%
RTG 1.1	Truck to stack	1948	1934	99.28
	Stack to truck	7655	7612	99.29
	Stack to stack	6051	6018	99.45
RTG 1.2	Truck to stack	1654	1651	99.82
	Stack to truck	8421	8382	99.54
	Stack to stack	6571	6551	99.70
RTG 2.1	Truck to stack	1697	1692	99.71
	Stack to truck	1612	1607	99.69
	Stack to stack	2340	2333	99.70
RTG 3.1	Truck to stack	58	58	100
	Stack to truck	235	233	99.15
	Stack to stack	1052	1036	98.48

4.6 Case: overhead crane work cycle modeling

4.6.1 Work cycle model training

Unlike in modeling the STS and the RTG cranes, at the time of compiling the results for this thesis, there were no data available from normal work of an overhead crane for work cycle model development. Therefore, a special experimental arrangement was made for an education and test crane presented in Section 2.5. The experimental test arrangement is descrbed in Figure 2.7. The initial position for the cycle was predefined so that each cycle would start from the same location. The task of the operator is to transfer the crane over an obstacle and then to catch a mass (1000 kg) at a predefined point. Another human operator buckles the mass to the hook of the crane. Once the mass is caught, the next task is to transfer it over another obstacle, place it carefully on the ground, and then another human operator releases the hook from the mass.

Based on the knowledge about the work objective, the work cycle was defined to follow the task hierarchy shown in Figure 4.7. Because the load is connected to the crane using a passive hook, which a human operator buckles up, only four main tasks could be reliably recognized. For a crane with an active grapple, a more sophisticated work cycle model could be developed. However, the task flow shown in Figure 4.7 provides a good starting point for analyzing the bottlenecks in the crane work.

For the model development, three operators completed the test race ten times. Thus, in total, 30 work cycles were available for training and validation of the model. During the test runs, a video was also recorded for training and validation purposes. The training data set of the HMM was constructed by labeling manually 15 work cycles (five from each operator).

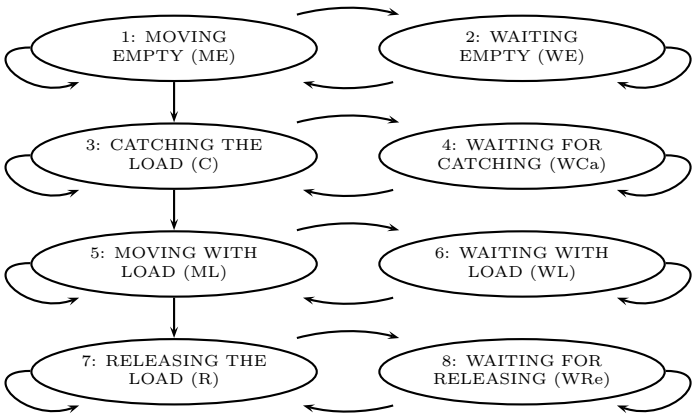


Figure 4.7: The task hierarchy, the most probable transitions of the overhead crane work cycle, and the associated task name abbreviations used in Table 4.6.

Then the HMM matrices were obtained by using the supervised training approach described in Section 4.2.3. The practical implementation of the training procedure was similar as described in the supervised training case of the simulation study Section 4.3.1 and in Appendix A.2.

4.6.2 Work cycle model validation

The model validation was performed by manually analyzing another set of work cycles which were compared against the state sequences given by the HMM-based work cycle model. The confusion matrix computed using (4.5) is shown in Table 4.6. According to the results, the HMM model manages to classify most of the states with sufficient accuracy. However, the same notions regarding to the reliability of the validation discussed in the STS case in Section 4.4.2 apply in the overhead crane case as well.

It is important to note that the overhead crane results reported in this thesis are only initial. The human operators in this experiment were not professionals, and the work site was artificial. Using data from work cycles of beginner operators might violate the repetitivity assumption of the work cycle modeling, because, according to the task time distributions computed based on the modeling results, significant amount of learning occurred during the data gathering period. Based on the lessons learned from the artificial test race, data were gathered also from an industrial crane. However, the results of that case are not reported in this thesis.

4.7 Discussion

The work cycle modeling method based on HMMs was presented. For the method to be applicable for modeling the work of an arbitrary machine, the work needs to fulfill the repetitivity assumption. That is, the successive work cycles (with same task types) needs to be similar. Work cycle model development, training and validation methods were described in terms of

Table 4.6: Time percentage confusion matrix of the overhead crane model based on the state time distribution (4.5). The abbreviations of the task names are given in Figure 4.7.

	ME	WE	C	WCa	ML	WL	R	WRe
ME	96	0	3	0	0	0	0	0
WE	1	99	0	0	0	0	0	0
C	2	0	87	2	9	0	0	0
WCa	0	0	4	94	0	2	0	0
ML	0	0	0	0	88	0	12	0
WL	0	0	0	0	0	100	0	0
R	0	0	0	0	0	0	99	1
WRe	0	0	0	0	0	0	2	98

the well-known basic problems of HMMs. In addition, a measure for recognition of anomalous work cycles based on the estimated task sequence was presented.

A simulation-based benchmark example was presented for testing different work cycle modeling methods. Based on the example, it was confirmed that if the work fulfills the repetitivity assumption, equally accurate results are obtained with supervised and unsupervised training of the HMM parameters in terms of the classification performance.

Because the human operator's work to be modeled is dynamic, the manual data labeling is a challenging task. In addition, the labeling is subjective. That is, the result depends on the person who performs the labeling. This is because for many tasks, it cannot be determined exactly when the task ends or begins, based on the available time series of the raw data and video material. Thus, for the task whose beginnings and ends are not trivial, the classification capability can be poor. One can try to increase the classification capability by defining new events which could separate those particular tasks. Nevertheless, the subjectivity of the manual data labeling has impact for the supervised training and the validation (for the models trained using either the supervised or unsupervised approach). In order to remove the uncertainty depending on the individual human analyst who labels the training data, one should use several independent analysts and then use the result averaged over all analysts in training (or validation).

Another method to improve the data labeling in the training phase could be to cluster the video data using the other known human intent recognition methods designed especially for video-type of input data. Although the video data are not available in the normal operation of the model, the initial labeling for the training could be done using that. Then the final HMM model could be trained using the supervised method and the labeled task sequences obtained from the video clustering. The clustering could be done using, for example, another HMM-based technique proposed in [88]. By using automated clustering which relies on the additional measurements such as video material, the data labeling for the HMM training can be carried out in an automatic manner, and the analyst-dependent uncertainty problem would not occur.

In this chapter, the work cycle modeling method was applied succesfully to three different

types of industrial cranes. In STS and RTG case, the unsupervised training method was used. In the overhead crane case, the supervised training method was chosen. In all cases, the HMM provided acceptable classification capability. In addition to the results reported in this thesis, the author was involved in development and implementation of the work cycle modeling approach for forest harvesters [156, 160, 159] and forwarders [118, 119]. The HMMs developed for forest machines are implemented to run online during normal work of the machines. Based on the three case examples reported in this thesis, as well as the previous work done in forest machines, it can be confirmed that the HMM-based work cycle modeling can be succesfully applied to various type of cranes.

5 Skill and performance evaluation of human operators

As described in Section 3.1, in the development of HAMC, the first objective is to recognize the human intent based on the available measurements. The intent recognition can be performed using several methods, but in this thesis, an HMM-based work cycle modeling is used as described in Chapter 4. In this thesis, the task sequence represents the objectives and goals of the human operator. Once the intent of the human operator has been recognized, the second task is to evaluate the skills of the human operator or alternatively model the actions of the human operator as a mathematical model of a controller. Whether to use the skill evaluation, human operator modeling or both depends on the interaction levels one desires to implement in the HAMC, as well as the available measurements. If the measurements do not contain enough information for human model identification, the human operator modeling is not possible. In order to implement the "skill evaluation" interaction level, obviously skill evaluation is needed. For implementation of the passive adaptation, one can alternatively use the skill evaluation method as a basis to determine the optimal parameters for the HMI, or the human operator modeling. In the former case, the passive adaptation becomes a reasoning problem and in the latter a controller tuning problem.

The objective of this chapter is to develop a skill evaluation method, which can be applied in the evaluation of the skill and performance of working machine operators. In order to provide a basis for human skill evaluation, one needs to understand the behavior of a human in working tasks. Therefore, let us consider the Rasmussen's Skill-Rule-Knowledge model for human performance shown in Figure 5.1. In the model, the human behavior is divided into three levels: skill-based, rule-based, and knowledge-based behaviors. The boundaries of the levels are not strict and typically while learning, some operations change from a higher level to a lower level behavior. The skill-based level represents the motor-sensory performance during actions which follow directly from the human intent. The skill-based behavior is executed mainly based on *signals*, which are sensory data representing dynamical time-domain variables. A human operator can process the signals directly as continuous variables. [127]

At the rule-based behavior level the human operator constructs a sequence of subroutines based on stored rules. The rules are based on experience of similar situations or derived based on continuous problem solving and planning. In contrast to the skill-based behavior level, which is mostly unconscious and the human operator cannot describe what information exactly is used to realize the behavior, the rule-based behavior is generally based on explicit know-how and the human operator is able to explain the rules to other persons. The rule-based behavior is executed mostly based on *signs*, which indicate the state of the environment with respect to the actions. Signs are features extracted based on the sensory input and the actions and they cannot be processed directly as continuous variables. Instead they activate stored patterns of rule-based behavior. [127]

The knowledge-based behavior level represents the human behavior in situations where no direct rule of control or know-how applies. In this level the human performance is goal-controlled, meaning that the goal needs to be explicitly formulated based on the analysis of the environment and overall objective of the work. Thereafter, a plan is developed by selection among several candidates which are analyzed before the optimum is chosen. The better the human opererator understands the work domain (causality and functions), the more succesful plan can

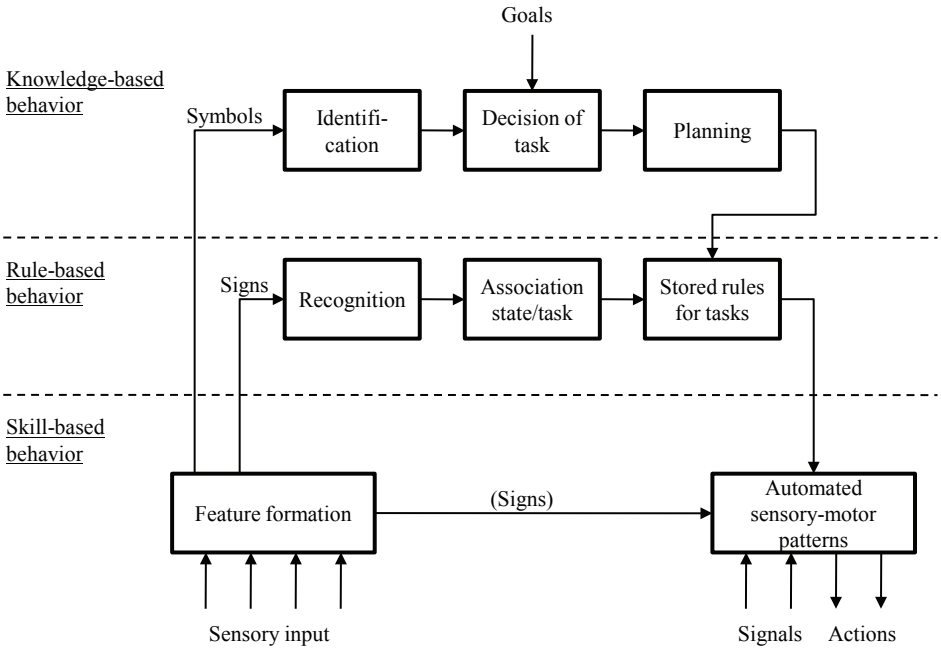


Figure 5.1: Illustration of the levels in human operator performance according to Rasmussen's Skill-Rule-Knowledge model. Adapted from [127]

be constructed. The knowledge-based behavior is based on *symbols*, which refer to information, variables, relations and properties of the system at hand. The symbols can be processed by reasoning or computation by the human operator. [127]

The objective of this chapter is to develop methods to evaluate the human operator's skills and performance in different levels of the Skill-Rule-Knowledge model based on the available measurement data. This chapter proposes a general framework for skill evaluation of human operators in mobile working machines. The chapter shows how the work cycle model can be used to derive skill indices by defining four frameworks to start with. Moreover, the chapter shows how the skill metrics can be further refined into skill components describing the skill to perform the work in several levels. The effects of external circumstances to the skill indices are clarified through an example, and a method for compensation is proposed. The chapter also describes how the skill evaluation and work cycle modeling system can be implemented in industrial applications. This chapter is mainly based on the author's publications [160] and [159].

5.1 Skill evaluation method

5.1.1 Overall description

The human performance can be defined as person's most likely outcome when performing a given task [167]. Thus, the skill can be interpreted as the average performance from several repetitions of a task. In addition, the skill level can be measured based on how smoothly a sequential process is executed [147]. In [146] the human skill in machine operation was defined as "an ability to manipulate machines accurately, fast, with high repeatability, and to cope with emergency circumstances". An excellent six-level hierarchy for skill evaluation based on the Rasmussen's model is proposed in [144]. However, based on the measurements available in the working machines, a simpler definition is made in this thesis. It is assumed that the skill of a human performing multi-objective complicated work with a working machine the is defined as follows.

Definition 1 *Human operator's skill in work performed with a mobile working machine consists of the following components: (i) an ability to control the machine, (ii) an ability to tune the control parameters of the machine appropriately to suit the operator's machine controlling skills, (iii) a knowledge of the work technique and strategy and (iv) an ability to plan and make decisions.*

With respect to the Skill-Rule-Knowledge model, only the first component represents "skill-based behavior". However, the term skill evaluation in this thesis refers to the general human performance covering all levels of the Skill-Rule-Knowledge model. The second skill component, an ability to tune and choose the control parameters suitable for one's skill level is a higher level component. The human operator needs to be able to analyze cause-and-effect relationships from the settings of the control parameters (speed, sensitivity, etc.) to the execution and realization of the control movements. This is related to the knowledge of the system "as a whole" and can thus be interpret as knowledge-based behavior. The third component, knowledge of the work technique and strategy relates basically to rule-based and knowledge-based levels. If the human operator chooses inappropriate rules for the current context, the poor performance is due to the rule-based behavior. If, on the other hand, the human operator constructs a rule which does not correspond to the causality in the real work scenario, the performance degradation is mostly due to knowledge-based behavior.

The structure of the skill evaluation system proposed in this thesis is shown on the left in Figure 5.2. At the lowest level there are the measurements available in the machine. The measurements are assumed to contain information about the control actions the human operator introduces to the system. In addition, the measurements might include the information about the prevailing operating conditions and the settings of the control interface. It would be desirable if the measurement set contains the information about the response of the system to the human operator's control actions. However, this is not always possible, especially in boom cranes because all joints do not typically have sensors for the measurement of the current state. In terms of the Skill-Rule-Knowledge model, the measurement level represents the sensory input for the human operator regarding the state of the system and the environmental conditions, as well as the actions of the human operator.

The second level in the skill evaluation system is the determination of the human operator's intent based on the measurements. In this thesis the intent (or plan) is interpret as the task or task sequence the human operator is executing. The task sequence recognition can be carried out by using the HMM-based work cycle modeling method described in Chapter 4. However, other methods can be used as well, depending the quality of the available measurements. The work cycle modeling provides the information of the starting and ending time of each task. This information can be used for association of other process measurements to the task. This is very important since good values of some measurements might depend strongly on the task at hand.[11] The task association allows to develop task specific skill indices. In terms of the Skill-Rule-Knowledge model, the task classification allows to evaluate the different control actions with respect to the executed task, which leads to "skill-based level" skill evaluation. In addition, the evaluation of task performance as well as the analysis of the executed tasks enable the evaluation of rule-based behavior. Analysis of the task sequence properties with respect to the operating conditions and the measurements as well as the decision making related tasks, enables the evaluation of the knowledge-based level skills.

At the third level, the effects of the prevailing operating conditions (sizes of objects, moving distances, task difficulties etc.) are compensated from the skill indices. Thereafter, the compensated values of the skill indices are evaluated statistically. That is, given the prevailing operating conditions, the difference between the current performance and the statistical reference (obtained from large amount of experimental data) is calculated. Finally, the skill indices can be aggeregated to describe the different areas of the human skill explained in Definition 1.

As an everyday-life example of the difference between a beginner and an expert in sequential task execution, consider the beginner and expert driving a car with manual transmission (example modified from [9]). The task sequence of the manual gear shifting process is illustrated on the middle and right diagrams in Figure 5.2. At the very early stage of learning the task sequence of the beginner is complicated. Each task needs to be cognitively rehearsed upon completion. When changing the gear, the beginner needs to check the location of the next gear before engaging the clutch. The beginner shifts the stick to the middle position but during this time, the location of the next gear may have been forgotten. So it needs to be rechecked. Moreover, the beginner might attempt to change to a wrong gear, and might face some other troubles as well. Thus, the beginner can execute only one task at a time and the task execution is hesitative and uncertain. Whereas the beginner executes task in series, the expert can perform parallel operations. The task sequence of the expert is significantly simpler, smoother, autonomous, faster and the execution very determined.

It should be emphasized that in the manual gear shifting, the driver should also perform other task necessary for driving such as monitor the traffic and other cars, use the gas pedal, and monitor the speed of the car. These tasks are done simultaneously with the gear shifting. Thus, the cognitive load of a beginner becomes very high. [159]

[11] For example, the good value of the average control level and the variance of the control signals for a boom crane depends on the task. If the task is to move an object for a long distance, the large control levels are natural, but when an object is picked with the crane the beginners tend to use small control levels with high variance.

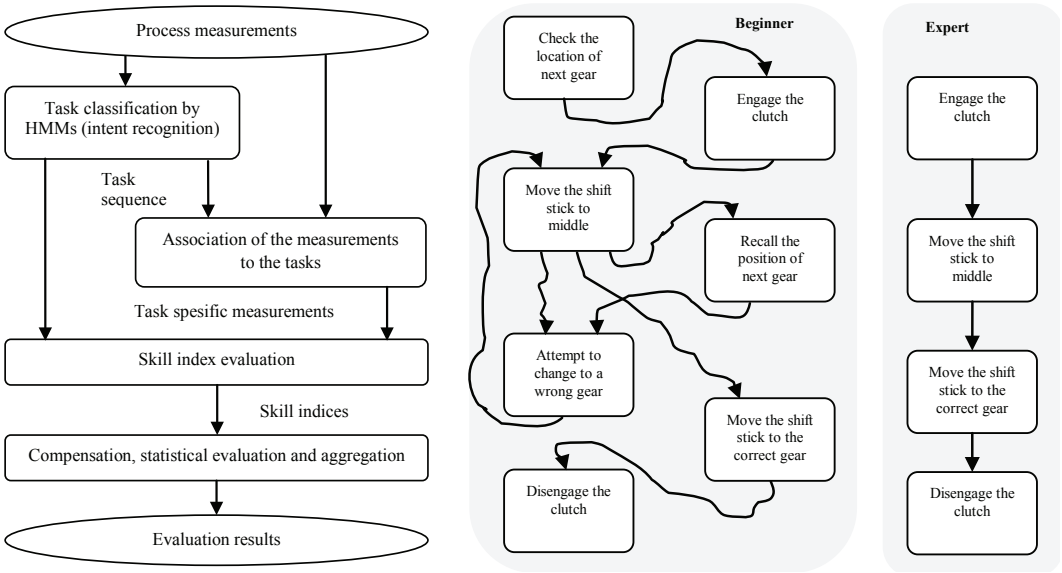


Figure 5.2: On the left, the skill and performance evaluation framework for human operated machines (adapted from [160]) and on the right, the illustration of the difference between beginner and expert in manual gear shifting process (adapted from [159]).

5.1.2 Illustrative example

To illustrate the idea of the HMM-based skill evaluation approach for machine work, let us consider an example using the overhead crane described in Section 2.5. The overhead crane data include the speeds of the motions in all three directions (hoist, trolley and gantry) as well as the position and load information. Using the raw data, the HMM-based work cycle model of the overhead crane was developed in Section 4.6. An example work cycle with the speed of the movements and the load information is shown in Figure 5.3. All variables in the time-series are scaled so that the maximum absolute values are one. In addition, the lower plot includes the task sequence corresponding to the raw data estimated using the Viterbi algorithm and the HMM trained in Section 4.6.

The result of the HMM-based work cycle recognition is the task sequence. In Figure 5.2 (left diagram) it corresponds to the second level from above. Once the task sequence is obtained, it is known at what time each task starts and ends. Thus, one can associate the measurements to the tasks (the third level from above in Figure 5.2). In the example work cycle shown in Figure 5.3 the association of the measurements to the tasks is illustrated by drawing vertical dashed lines correponding the time instants the task is changed (according to the trained HMM). Now one can easily note that, for example the speed levels of the movement directions differ between the tasks. This infomation can now be exploited in the skill evaluation because one can extract task-specific features from the raw data measurements. In the following sections this overhead crane example is used to illustrate the idea of the skill index calculations.

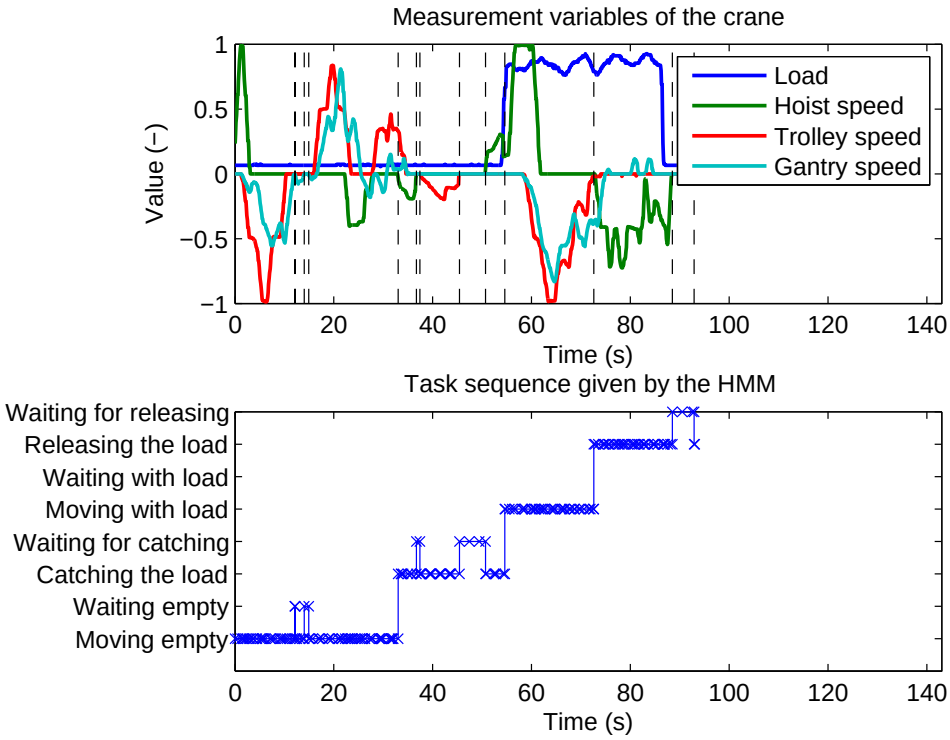


Figure 5.3: An example of association of the measurements to the tasks. A cycle executed using the crane described in Section 2.5 and the task sequence is estimated using the HMM developed in Section 4.6. In the upper plot, the task change times are illustrated by dashed vertical lines.

5.2 Development of skill indices using task sequence data

5.2.1 Task efficiency based indices

In the task efficiency approach it is assumed that a skilled operator accomplishes a single task, or a sequence of tasks efficiently without wasting resources. The resource here can refer to time (as in [106, 132, 131]) but in general any resource can be considered.

Let R_t denote the amount or rate the resources are consumed between the observations at t and $t+1$. Thus, the total efficiency E_i of the task S_i during the task sequence within T is obtained by summation over all observations as

$$E_i = \sum_{t=1}^{T} \delta_{i,q_t} R_t, \tag{5.1}$$

where δ_{i,q_t} is the Kronecker delta[12], which describes wheter two arguments are equal or not. Moreover, the average resource consumption of a single task in the task sequence can be computed as

$$E_i = \frac{\sum_{t=1}^{T} \delta_{i,q_t} R_t}{\sum_{t=1}^{T} \delta_{i,q_t}}, \tag{5.2}$$

which can obviously be computed only if at least one transition to the state S_i appears. Depending on the task definitions, the task efficiency as the skill measure reflects mostly to the skill component *(i)*[13] but can reflect to all components *(i)–(iv)* of Definition 1. Similarly, in terms of the Skill-Rule-Knowledge model, the human behaviors in all three levels affect to the task efficiency.

As an example for the task efficiency as a skill measure let us consider the data shown in Figure 5.3. The load catching time is a difficult task which, in the overhead crane case describes the skills of the operator and the person buckling the load. For catching time, one should consider the states "Catching the load" and "Waiting for catching", where the waiting is mostly due to the assisting person connecting the load. Thus, $i = \{3, 4\}$ and R_t is defined as time consumption during the task. In the example work cycle, the skill variable E_i describing now the total catching time becomes $E_{3,4} = 22.3$ seconds.

5.2.2 Task sequence complexity based indices

In the task complexity approach it is assumed that a skilled operator accomplishes the work objective with a fairly simple sequence of tasks. In the gear shifting example, the expert used four or less distinct tasks to achieve the same goal as the beginner, who used at least seven tasks. Moreover, the beginner might execute some tasks more than once.

At the simplest, the complexity of the task sequence can be calculated by letting f_i to denote the number of transitions to the state S_i from any other state, and summing over the sequence of tasks as follows.

$$f_i = \sum_{t=2}^{T} \delta_{i,q_t} \left(1 - \delta_{i,q_{t-1}}\right) \tag{5.3}$$

Now, f_i describes how many times the task S_i was executed during the task sequence. In order to measure the complexity of the whole task sequence one can sum up the single task execution indices f_i to obtain

$$f_{tot} = \sum_{i=1}^{N} f_i, \tag{5.4}$$

where N denotes the number of possible tasks (states of the HMM). The complexity approach measures the skill components *(iii)* and *(iv)* depending on the definition of f_i. In terms of the Skill-Rule-Knowledge model, the task sequence complexity measures mostly the rule-based and knowledge-based behavior. Applying a wrong rule for a work situation, which basically represents a misrecognition of the situation, might lead to re-execution of certain tasks. On the

[12]Two argument Kronecked delta is $\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$.

[13]With the assumption that a human operator performs the work smoothly, autonomously and without hesitation, as it is done in the skill-based level of the Skill-Rule-Knowledge model.

other hand, if the human operator has wrong "mental model" of the work and the associated causality relationships, the rules themselves might be wrong although they are correctly applied. In such case, the task sequence complexity measures the knowledge-based level.

Again, let us consider the overhead crane example data shown in Figure 5.3. Using the example data, the complexity of the task sequence $f_{tot} = 13$. Here, it should be noted that if the recognized task sequence is not smooth due to, for example noise or anomalous work, the complexity of the task sequence will be high. However, as it will be discussed later in Section 5.4, in the practical implementation one should always evaluate longer time periods, not only one or two work cycles. Thus the performance variation due to immeasurable noise can be averaged.

5.2.3 Planning and decision making ability based indices

A human operator working in a complex and dynamic environment with time pressures due to the work completion, an expert typically chooses the first available solution without spending time for considering the optimum choice among different alternatives [79]. Once the decision is made, the task is executed determinedly. Thus, an expert can quickly make a succesful decision and the tasks related to decision making need to be executed only once. On the contrary, a beginner has to perform tasks related to decision making several times upon completion of the work objective. Several direction changes, stoppages, or waiting times in the trajectory of the movement may indicate poor decision making skills.

The planning and decision making approach can be utilized mathematically analogously to the complexity approach by using (5.3), where f_i describes the number of transitions to the decision making related task S_i of interest. The number of direction changes or stoppages during a task can be evaluated using (5.1) by defining R_t as the particular value of interest during the task. The planning and decision making approach measures mainly the skill component *(iv)* but it also can measure the skill component *(i)*. In terms of the Skill-Rule-Knowledge model the planning and decision making approach measures mostly the rule-based and knowledge-based levels, but can also be affected by the skill-based behavior. Direction changes or stoppages during the movement may indicate an unsuccesful plan to cope with an unexpected situation (knowledge-based level) or the application of an inappropriate rule for the situation (rule-based level). High motor-sensory skills reflect to the smoothness of the operation [127]. Thus, direction changes, stoppages or high variance of the control signal may indicate the low machine control skills. Thus, this approach can also be used to measure the skill-based level.

In order to illustrate the planning and decision making ability as well as the task difficulty based indices, three new features were extracted from the measurement data shown in Figure 5.3: number of active movements, direction changes, and the resultant speed. The time-series of the new features are shown in Figure 5.4. A movement is defined to be active if the speed of the corresponding degree of freedom is greater than 0.15. The direction change (for any movement degree of freedom) is defined when the corresponding movement velocity changes from smaller than -0.15 to greater than 0.15 or vice versa. The resultant speed is obtained by square root of the sum of squared velocities in each degree of freedom.

The overhead crane work performed using the test crane did not, in practice include planning. The task was simply to transfer the mass from a point to another. In normal work of such crane, for example if the task is to transfer several different objects, planning of the work is

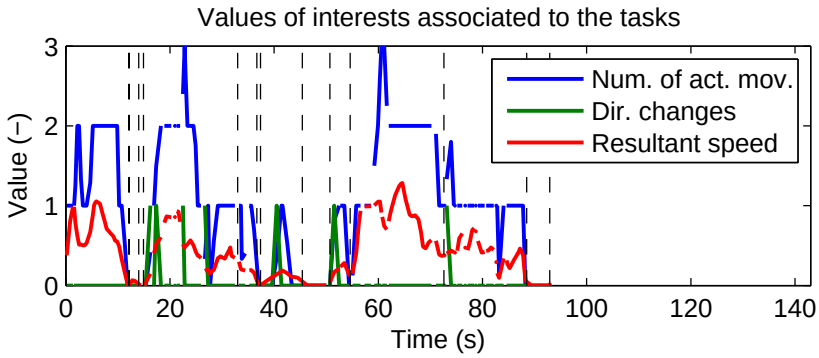


Figure 5.4: An example calculation of values of interests for task-specific skill indices based on the data shown in Figure 5.3.

required from the operator. Despite the lack of practical planning work in the example data set, in order to illustrate the computation of the indices described above, let us consider the number of direction changes during "Moving empty" task (task index 1). Define f_i as the number of direction changes during "Moving empty". The summation of the direction change flags shown in Figure 5.4 gives $f_1 = 4$. Since the work in the demo crane case did not include planning, this value now mostly measures the operator's machine controlling skills.

5.2.4 Task difficulty based indices

According to the Fitts' law the relationship between the task difficulty index and the task completion time, where the index of difficulty is defined based on the ratio between the total movement distance and the target size is a good measure of the human skill [39]. However, in mobile working machines the task difficulty, in terms of the Fitts' law, cannot typically be directly measured. Instead, it can be assumed that a motivated operator utilizes fully his/her execution capacity. In other words, an expert can perform a task or a sequence of tasks in an efficient manner but with the cost of high difficulty (e.g. using several parallel control functions in overlapping sequences) whereas a beginner needs to use easier and less efficient solutions (e.g. controlling machine functions separately in series). In [127], the skill-based behavior was referred to as smooth, autonomous, and highly integrated actions. Thus, one can assume that the average difficulty of the task (e.g. as parallel controls) can be used as an indirect measure of the human operator's machine controlling skills.

The work cycle recognition enables the association of the process measurements to the tasks as seen in Figure 5.2. Thus, the human operator's control signal characteristics can be studied task-wise by using the signals which measure the handling of the control levers.

A task difficulty-based skill index for task S_i can be obtained using (5.2) by defining R_t as the value of interest (mean control rate or number of parallel controlled functions) during the task S_i. The task difficulty approach measures the skill component *(i)*. Together with the task efficiency metrics (5.1) and the knowledge about the current parameter setup of the control

system the task difficulty approach can be used to measure also the skill component *(ii)*. The task difficulty-based indices measure mostly the skill-based behavior.

As an example, let us consider the overhead crane data features shown in Figure 5.4. For the task difficulty based indices one can choose, for example the average number of active movements and the average resultant speed during the moving tasks. The average number of active movements during "Moving empty" is 1.31 and during "Moving with load" 1.67. The average resultant speed during "Moving empty" is 0.51 and during "Moving with load" 0.78.

5.3 External circumstances and operating point

When working with a machine in changing operating conditions the performance variations caused by the natural external conditions can be significantly greater than the variations caused by variations in machine's technical performance or the variations caused by the human operators' skill differences. Similarly as in the machine work, in process industry the operating conditions need to be considered in the human operator evaluation as it was done in the mineral processing study reported in [123].

Consider yourself being asked a question "Is $-2°$C cold in Helsinki?", where you would answer something like "It depends on the time of the year" [40]. Similarly in machine work, the performance of a task depends on the circumstances the task is executed. For example, the time used for moving an object from point to point by a crane depends on the physical dimensions of the object and the distance the object is moved. Therefore, before comparing the performance of different realizations of the moving tasks, one should first study the circumstances the task was executed in, and then the evaluation should be done with respect to those circumstances. The external circumstances, from here on, are referred as the *operating point*, which includes all measurable environmental variables which affect to the performance of the machine and the operator.

The operating point compensation can be performed by conditional evaluation. In the simplest case, one can assume that the performance variable x is conditionally Gaussian. If the vector $\boldsymbol{\zeta}$ denotes the operating point (each component describes the value of an environmental variable) the conditional expectation of the performance variable is

$$\bar{x}_\zeta = \mathrm{E}\left(x|\boldsymbol{\zeta}\right) \tag{5.5}$$

and the conditional standard deviation

$$s_\zeta = \sqrt{\mathrm{VAR}\left(x|\boldsymbol{\zeta}\right)}. \tag{5.6}$$

The compensated performance variable value x_ζ can now be obtained simply by standardization

$$x_\zeta = \frac{x - \bar{x}_\zeta}{s_\zeta}. \tag{5.7}$$

If the conditional Gaussianity is not a valid assumption, one can use another statistical metrics, such as percentiles to measure the normal variation around the operating point. The idea for operating point compensation by binning the metrics according to a predefined classes and then

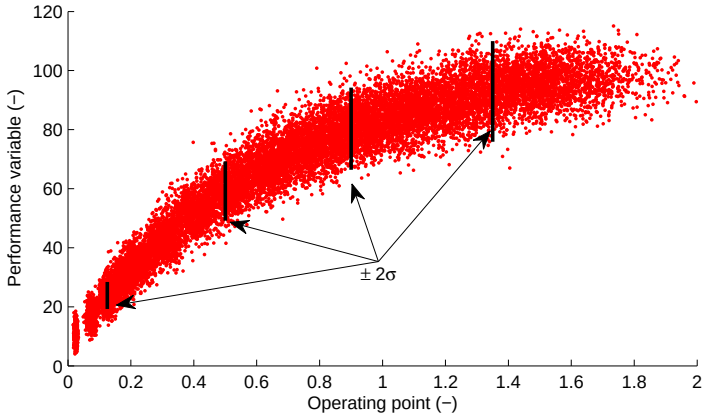


Figure 5.5: Example data describing a possible relationship between the performance variable and the operating point.

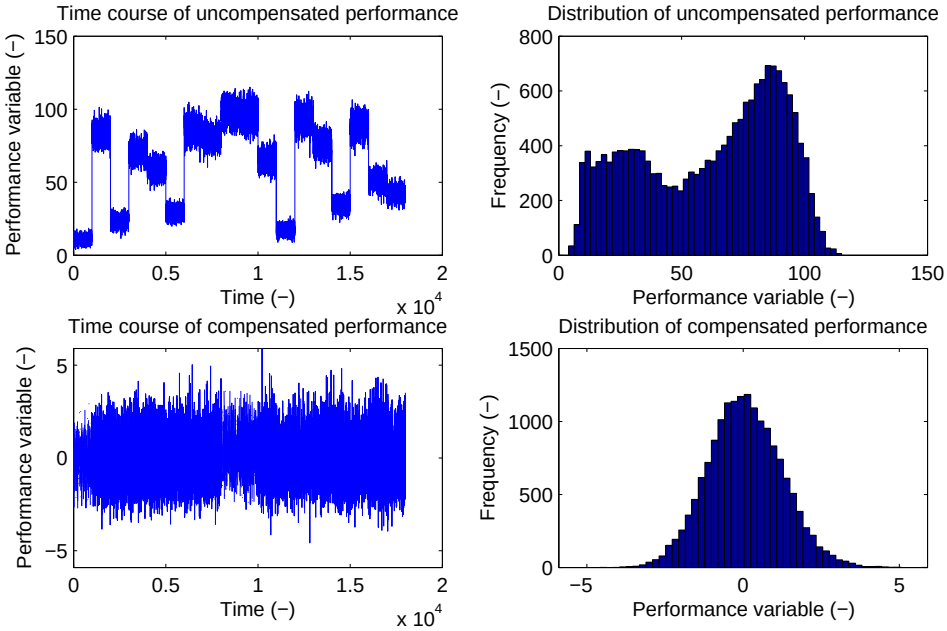


Figure 5.6: The time course and the distribution of the data seen in Figure 5.5 before compensation (upper) and after compensation (lower).

using the conditional evaluation for normalized performance indices was introduced in [57].

As an example of the conditional evaluation, consider the relationship between the performance variable x and the operating point $\boldsymbol{\zeta}$ seen in Figure 5.5. The data set is generated according to the parameters obtained from the true performance data of forest harvesters (several hundred

thousand samples) and then assuming conditional Gaussianity[14]. One can see that mean and the variance of the performance variable depend strongly on the operating point.

In Figure 5.6 the upper plots show the time course and the distribution of the performance variable before the compensation. One can see that the variation around the operating point is significantly smaller than the difference between the average values of the operating point changes. The resulting histogram describing the distribution is thus multimodal. Then the conditional standardization (5.7) was used for each data point separately. Now the time course seems like a random signal, exactly as it should be, because the data were generated by using the Gaussian random number generator. The distribution of the compensated performance variable is unimodal and follows the shape of the Gaussian probability density function.

5.4 Statistical learner model

The idea of the statistical learner model was originally proposed by Lauri Palmroth in [119] where it was used to determine the skill levels of operators for an intelligent coaching system. The author of this thesis was involved mostly in the practical implementation, especially for skill evaluation of harvester opererators.

The human performance has several immeasurable sources of uncertainty. The concept of the statistical learner model was developed to account for the randomness associated with the performance of a man-machine process. It assumes the performance variations due to the randomness in human motor-sensory processes, emotions, fatigue, and motivation as noise, which can be averaged away by using a long enough time window. That is, for each learner there exists the most likely performance level, which can be interpreted as the average performance from several repetitions of a certain task. This is important, since the human factors cannot be measured in real-life industrial processes.

The most likely level of a skill metric is calculated using a long enough averaging time window. It is assumed in the statistical learner model that there exists a level of high performance (extreme value) that is very rarely exceeded. The most likely performance levels of the operators are assumed to be distributed according to a Generalized Extreme Value Distribution (GEV). If x denotes the averaged value of a performance index of an operator during a time window, say, one work shift (roughly eight hours). The probability density function of a GEV is [85]

$$p_X(x) = \frac{1}{\sigma}\left(1+z\right)^{-1-\frac{1}{\xi}} e^{-\left(1+z\right)^{-\frac{1}{\xi}}}, \tag{5.8}$$

where

$$z = \xi\left(\frac{x-\mu}{\sigma}\right), \tag{5.9}$$

and ξ, μ and σ are the parameters that determine the shape of the distribution. The statistical performance level of the operator (the eight hour shift during which x was recorded) can be obtained from the cumulative distribution function

$$\mathrm{P_r}\left(X \leq x\right) = e^{-\left(1+z\right)^{-\frac{1}{\xi}}} \tag{5.10}$$

[14]This is not exactly a valid assumption here but it is assumed for simplicity.

if large values of x correspond to the desirable performance. If small values of x signify the desirable performance, the complement distribution

$$\mathrm{P_r}\left(X > x\right) = 1 - \mathrm{P_r}\left(X \leq x\right) \tag{5.11}$$

is used.

Depending on the sign of the shape parameter there are basically three types of GEV distributions. In I-type (Gumbel) the shape parameter $\xi = 0$, and the probability density function is obtained as $\lim_{\xi \to 0}$ of (5.8), which results in

$$p_X(x) = \frac{1}{\sigma} e^{\left(-e^{-(x-\mu)/\sigma} - \frac{(x-\mu)}{\sigma}\right)}. \tag{5.12}$$

The cumulative distribution function of the Gumbel-type distribution is obtained as $\lim_{\xi \to 0}$ of (5.10), which results in

$$\mathrm{P_r}(X \leq x) = e^{-e^{-(x-\mu)/\sigma}}. \tag{5.13}$$

As for the other types of GEVs, the type II (Frechet) GEV is obtained when $\xi > 0$, and the type III (Weibull) when $\xi < 0$.

The compensation and the statistical learner model is illustrated with an imaginary example in Figure 5.7. Now, x is the performance variable whose high values signify good performance. The plot on the upper left describes the value of the performance variable as a function of the operating point ζ. In the data, there are 18 operating points whose average values range from 0.025 up to 1.55. For each of the six operators (indicated by different colors), 20 points are generated around each operating point. The points are generated from a GEV distribution whose parameters μ and σ are varied around each operating point for each operator.

The middle plot in the upper row describes the time course of x for each operator. From the uncompensated values it is difficult to recognize good and bad operators. Then the values of x are compensated according to the distribution of the data seen in Figure 5.5. Although the parameters in (5.7) are now from the Gaussian distribution and x is distributed according to a GEV, in this example they provide a reasonable standardization for the performance variable.

The rightmost plot on the right describes the time course of the compensated performance variable x_ζ. Now one can easily spot that the blue operator (OP1) has the worst and the magenta (OP6) the best performance. In the statistical learner model the randomness associated with the human performance is compensated by considering the average performance during a long enough time window. In this example, 20 points were generated around each of the 18 operating points. Thus altogether there are 360 data points for each operator.

For the evaluation, the distribution of x_ζ was determined. The histogram describing the distribution of is seen in the lower left plot. A Gumbel type of a GEV was fitted to x_ζ data by using the maximum likelihood method. In the lower middle plot one can see the positions of each operator in the probability density distribution describing the average performance levels of each operator. In the lower right plot one can see the evaluations given by the average compensated performance of each operator.

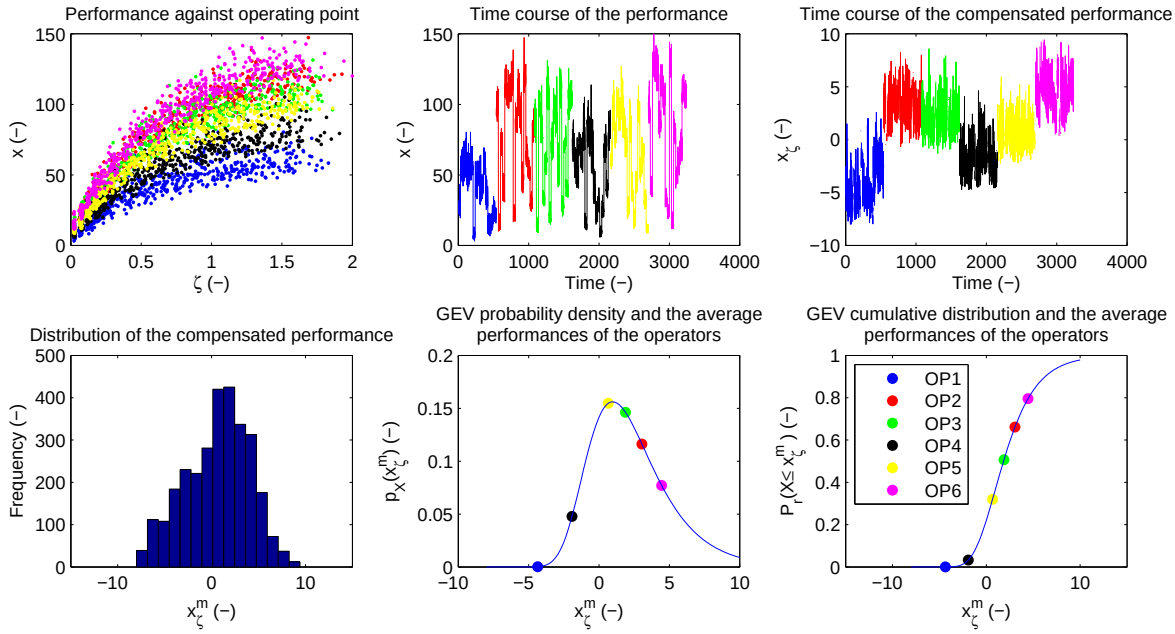


Figure 5.7: An imaginary illustration of the operating point compensation and the statistical learner model.

5.5 Skill index hierarchy

In the author's publication [160] a hierarchical FIS-based evaluation system was proposed to aggregate the low level evaluations into the higher level task specific skill component evaluations. The structure of the hierarchical evaluation system is shown in Figure 5.8. In this thesis, the index hierachy is as presented in Figure 5.8, but the aggregation is performed by using simple weighted averages as well as "AND" operators, instead of full FISs.

The index hierarchy for the skill evaluation can be determined as follows. In the measurement level the skill indices are obtained from the statistical learner model. At this stage, the high values of the skill indices signify good performance. At the task skill level, the measurement level indices are combined into task-specific evaluations for all skill components *(i)-(iv)*, if possible. Each local task-specific evaluation provides index values for all skill components. In the final phase, the local skill component evaluations are combined into global indices describing the overall levels of the components. Here, one should note that the skill index hierarchy is not designed to construct the same hierarchy as the behavioral levels in the Skill-Rule-Knowledge model in Figure 5.1. The aim here is simply to refine indicators to measure the different components of skills based on several low-level skill indices. This is done because the available measurements in the normal work of the machines allow only development of indirect skill indices. Generally, in terms of derivation of the skill indices, the planning and decision making as well as the work technique and strategy needs higher level reasoning than the evaluation of pure machine controlling skills. However, the interpretation of the indices

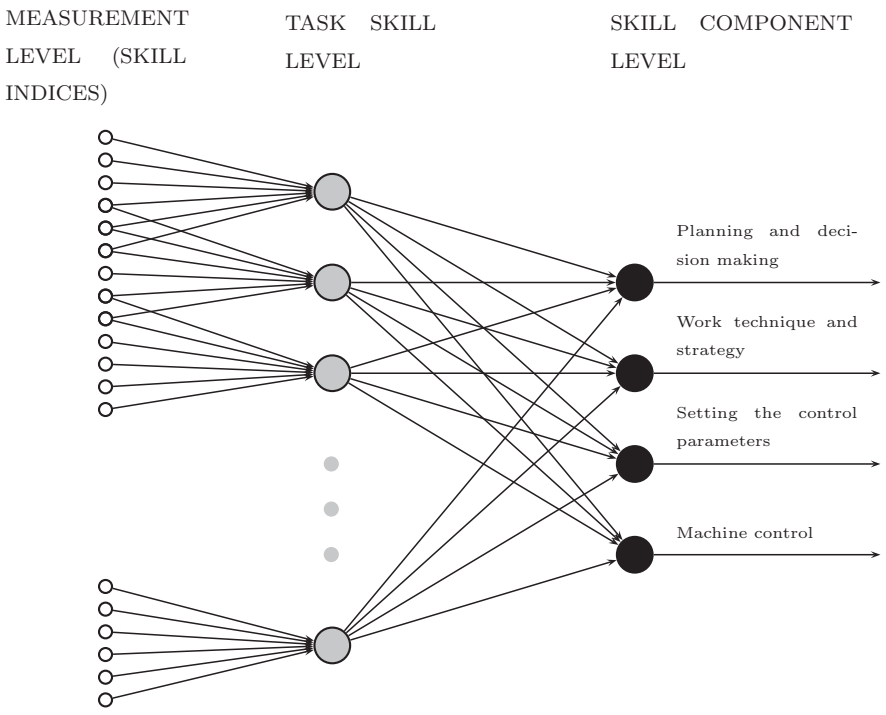


Figure 5.8: The skill index hierarchy: from measurement level to skill component level.

and the implementation of the required reasoning systems are application dependent and they are not discussed further in this thesis.

The higher level skill indices can be formed from the lower level indices using several methods. The simplest method is to use the weighted sum of the indicators as

$$I^{(k)} = \frac{\sum_{i=1}^{N} w_i I_i^{(k-1)}}{\sum_{i=1}^{N} w_i}, \tag{5.14}$$

where I^k $(0 \leq I^k \leq 1)$ describes the value of the desired skill component evaluation at the level k $(1 \leq k \leq 3$ because there are three levels in Figure 5.8), w_i the relative importance of that index for the skill component evaluation, $I_i^{(k-1)}$ $(0 \leq I_i^{(k-1)} \leq 1)$ the value of the skill index i at the level $k-1$ which evaluates the corresponding skill component, and N the number of indices for that particular component in level $k-1$. Alternatively, one can use AND-type of aggregation, that is, the product of two indices. For example, in the Fitts' law there are a realationship between the task completion time and the index of difficulty (defined based on the movement distance and the target size). Similar relationships can be produced using

Skill measure	Value	Evaluation	Weight
Moving empty time	31.3	0.7	2
Moving with load time	18.3	0.8	2
Num. of act. mov. (Moving empty)	1.31	0.66	4
Num. of act. mov. (Moving with load)	1.67	0.84	4
Dir. changes (Moving empty)	4	0.30	1
Res. speed (Moving empty)	0.51	0.51	2
Res. speed (Moving with load)	0.78	0.78	2

Table 5.1: Illustration of the measurement level of the skill index hierarchy in the overhead crane case.

Task	Machine control skill using (5.14)	Machine control skill using (5.15)	Weight
Moving empty	0.60	$\sqrt{\frac{4\cdot 0.66+1\cdot 0.3+2\cdot 0.51}{4+1+2}\cdot 0.7} = 0.63$	1
Moving with load	0.82	$\sqrt{\frac{4\cdot 0.84+2\cdot 0.78}{4+2}\cdot 0.8} = 0.83$	1

Table 5.2: Illustration of the task skill level of the skill index hierarchy obtained using the "weighted sum" and "AND" approaches in the overhead crane case.

AND-type of rules, which can be formed simply as

$$I^{(k)} = \sqrt{I_N^{(k-1)} \cdot \frac{\sum_{i=1}^{N-1} w_i I_i^{(k-1)}}{\sum_{i=1}^{N-1} w_i}}, \tag{5.15}$$

where the N^{th} skill index is now used as the AND-condition. Here, $I_N^{(k-1)}$ could for example be the evaluation of the task execution time. The square root is used to produce evaluations which favor high values.

There are several advantages of using the hierarchical approach for skill evaluation. Since the work performed by mobile working machines is multiobjective, it is difficult to find a single index to describe skills at the measurement level. Therefore, the number of the skill indices at the measurement level is high. When using the hierarchical approach, it is easier for the operator to monitor his/her performance from smaller number of indices. If the evaluations in some component are poor, the operator or a teacher can proceed to the task level and analyze more carefully which tasks need to be trained.

To illustrate the idea of the skill index hierarchy, let us consider the overhead crane example described previously. In particular, let us try to derive the machine controlling skill index for the overhead crane operator based on the measurements shown in Figure 5.3. At the measurement level, the skill indices are evaluated with respect to the operating point using the

statistical learner model. In this example, the evaluations are completely heuristic and they are not based on real statistical distributions. The purpose of this example is to illustrate how several indirect measurements of different aspects of human skill can be combined to indicate the skill components at the task level and at the overall skill component evaluation level.

At the measurement level, the indices describing the machine controlling skills in the overhead crane example were described in Section 5.2.3 and Section 5.2.4. In addition to the planning and decision making, the number of direction changes in the movement trajectory can imply how well the operator is able to control the different movement directions. In Section 5.2.3 the number of direction changes during "Moving empty" task was $f_1 = 4$. In Section 5.2.4, the average number of active movements and the average resultant speed during "Moving empty" and "Moving with load" tasks were computed. The values for the measurement level skill metrics are shown in Table 5.1. The table also shows values for "Moving empty" and "Moving with load" times, which were not shown in the previous text. In addition, the table shows the evaluations for the associated skill metrics scaled to the range $[0, 1]$, where 1 signifies a desirable value. As discussed above, the scaling used shown in Table 5.1 is purely heuristic. In the rightmost column of Table 5.1, the relative importances of each measurement level skill indices are shown. The weights here are only illustrative, but in general, the number of parallel movements during task execution can be considered as a good skill measure [5, 96, 127]. Therefore, the number of active movements describing the parallel control ability of the operator is given weight four. The average resultant speed is given weight two and the number of direction changes is given unity weight. The task execution times are given weight two.

By using the weights and the weighted sum aggregation (5.14) or alternatively the AND-aggregation (5.15), the task level skill component evaluations for the machine controlling skills are obtained. The results are shown in Table 5.2. Finally, using the weights shown in the table, the overall machine control skill evaluation can be obtained as $(1 \cdot 0.60 + 1 \cdot 0.82)/(1 + 1) \approx 0.71$ (from weighted sum values) or $(1 \cdot 0.63 + 1 \cdot 0.83)/(1 + 1) \approx 0.73$.

5.6 Case: skill evaluation of forest harvester operators

5.6.1 Setting up the experiment

In [160], the skill evaluation method was used to evaluate in total nine operators of two separate harvesters. The operating conditions and the operators of the machines were known. In addition, part of the work was filmed on a video so that the skill evaluation results could be confirmed. In this thesis, the analysis is extended for ten operators of three harvesters. In general, the operating conditions or the operators are not known, but the operating point compensation method is used to obtain comparable results for the evaluation.

The operators under evaluation are shown in Table 5.3. In addition, the associated machines, numbers of work shifts for the operator, and the total numbers of stems for the operators are shown. The operators OP1, OP2 and OP3 are from the first harvester, operators from OP4 to OP8 from the second, and the operators OP9 and OP10 from the third harvester. For the operators OP1, OP4, OP5, and OP6 there were data available from significant amount of work shifts. At the maximum, 21 work shifts were accepted from one operator for the analysis.

The parameters of the statistical learner model, and the operating point compensation models

Table 5.3: Operators, harvesters, amount of work shifts, and total amount of stems processed.

Operator	Harvester	Work shifts	Stems
OP1	1	21	11228
OP2	1	7	2434
OP3	1	1	220
OP4	2	21	9619
OP5	2	18	6004
OP6	2	16	5291
OP7	2	2	701
OP8	2	2	1040
OP9	3	3	752
OP10	3	2	570

were trained by using data from six harvesters. In total, the data set contained roughly 200000 samples (stems processed). For each harvester 200 sample average was calcultated. By using an averaging window of 100 stems, the size of the training data set was roughly 2000 samples. These data were used to train the compensation model and the statistical learner model for each performance variable.

The skill and performance variables for the harvester operator analysis are shown in Table 5.4. In addition, the table shows the equations used to calculate the skill index, and the weights which are later used for aggregation in the skill index hierarchy. The weights are chosen by hand based on the author's comprehension about the reliability of the particular variable as a skill metric. In total, there are 11 variables which describe the skill components *(i)*, *(iii)*, and *(iv)*. The skill component *(ii)* was not evaluated because the information of the current parameter setup was not available in the data set. In addition to the 11 skill variables describing different task-specific skill components, the overall productivity V1 is used to rate the operators.

5.6.2 Testing the difference between the operators

In order for the skill indices to be reliable, the levels of the index values should show variance between the operators. Since the indices are not normally distributed, the standard analysis of variance cannot be used. Instead, the differences between the medians of the compensated performance indices can be tested using the Kruskal-Wallis [45]. The Kurskal-Wallis test is a conventional nonparametric method to test whether median values in two or more groups are statistically different. The null hypothesis in this case is H_0 : "The performance levels of the operators are equal". For each operator, the average performance data were grouped so that one group consisted of 50 samples (processed stems). The number of groups for each operator can be seen in Table 5.5.

The Kruskal-Wallis test was performed to test the null hypothesis. As a rather trivial result, the null hypothesis can be rejected for each compensated performance index with any level of significance. For the Kruskal-Wallis test, it is enough for rejecting the null hypothesis that at

Variable	Description	Eq.	Sc.	Weight
V1	Overall productivity	-	-	-
V2	Main task 1 time	(5.1)	*(iii)*, *(iv)*	1.0
V3	Main task 1 parallel control	(5.2)	*(i)*	1.0
V4	Main task 2 time	(5.1)	*(iii)*, *(iv)*	1.0
V5	Main task 2 parallel control	(5.2)	*(i)*	1.0
V6	Main task 3 strategy 1	(5.1)	*(iii)*	0.4
V7	Main task 3 parallel control	(5.2)	*(i)*	0.4
V8	Main task 3 strategy 2	(5.1)	*(iii)*	0.4
V9	Main task 4 decision making	(5.1)	*(iv)*	1.0
V10	Main task 4 decision success	(5.1)	*(iii)*, *(iv)*	1.0
V11	Main task 2 complexity	(5.3)	*(iii)*, *(iv)*	0.5
V12	Main task 3 complexity	(5.3)	*(iii)*, *(iv)*	0.5

Table 5.4: Skill and performance variables, the related equations, the associated skill components, and the weights for aggregation.

least one operator's median differs from others [123]. Nevertheless, the box plot describing the distribution operator-wise of the 50-sample data is shown in Figure 5.9. The notched boxes represent the uncertainty estimates of the medians (circles with dots in the middle), and the whiskers the Interquartile Range (IQR). It is easy to note that the medians of the 50-sample averages differ clearly between the operators, although, some variables have outliers (denoted by circles outside the whiskers).

The test becomes more meaningful when pairwise tests are performed for each operator and the variable. The pairwise analysis was performed so that for each operator pair i, j $(i \neq j)$ the hypothesis H_0 : "The medians of the operators are equal" was tested. The operator OP3 was neglegted from the tests because the number of groups shown in Table 5.5 is too low. Thus, for each variable eight tests were performed in total.

After the tests, the number of test results with the resulting p-value smaller than 0.03 were counted. In other words, the number of cases where the null hypothesis is rejected in 3 % level of significance. The results are shown in Table 5.6. Previously, in the discussion of Table 5.4, it was claimed that variables V6, V7 and V8 are the most unreliable. From the "Percentage" row of Table 5.6 it can be confirmed that especially variables V7 and V8 have the lowest number of testes where the null hypothesis is rejected. For the other variables, one can conclude that there are at least minor differences between the operator-wise medians.

It must be remembered that only the effect of the variation in tree size was taken into account in the operating point compensation method for the variables. For most variables, also the tree species, and stand properties should be accounted for. However, because there were so few data available, where the latest HMM-based work cycle recognition is implemented, the tree species could not be accounted for. The information of the stand properties were not available. In addition, the technical performance variations can have, at least, a small impact on the skill variables. These were not considered in the skill evaluation, but are studied more closely later

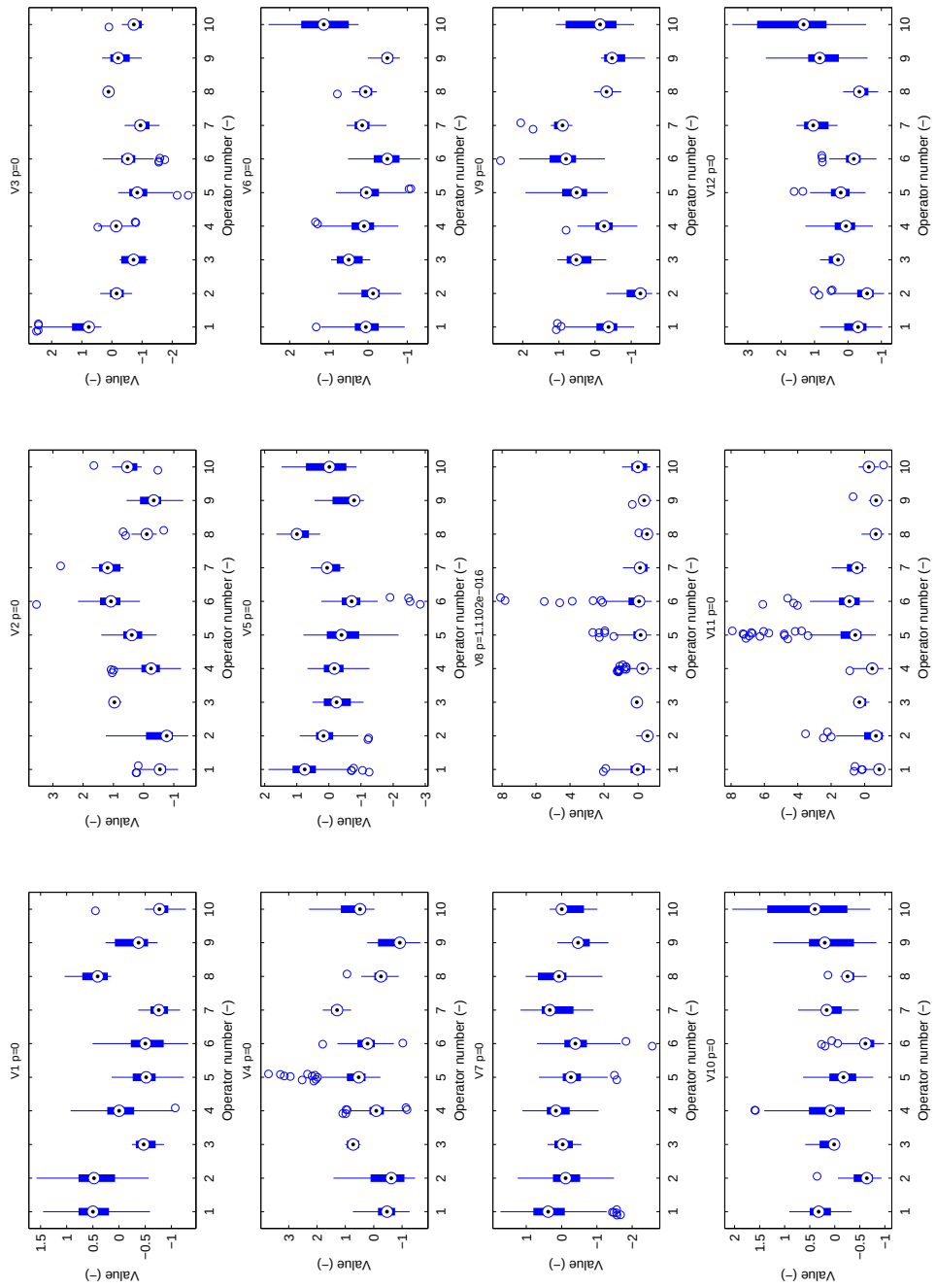


Figure 5.9: The box plots decribing the distributions of the skill evaluation variables for each operator. A circle with a dot in the middle is the median.

Table 5.5: Number of 50-sample average groups for each operator.

Operator	Number of groups
OP1	215
OP2	44
OP3	4
OP4	183
OP5	110
OP6	98
OP7	13
OP8	19
OP9	14
OP10	10

in this chapter.

5.6.3 Statistical learner model illustration

Based on the roughly 2000 samples of the 100 sample averaged data, the compensation model and the statistical learner model parameters were trained for each skill variable. The parameters of the operating point compensation model were fitted by binning the samples into boxes according to the compensation variable, and then the normalization parameters were obtained as the median ($\bar{x}_\zeta$ in (5.7)) and the interquadratile range (s_ζ in (5.7)). The parameters of the GEV distributions were fitted by using the maximum likelihood method implemented in Matlab Statistics Toolbox function "gevfit".

The illustration of the operating point compensation and the use of the statistical learner model is shown in Figure 5.10 for the variable V1 (Overall productivity). In the upper left the average performance x of each operator is plotted against the operating point ζ. In addition, the distribution lines of the operating point compensation model are shown. It can be seen that the operators have worked in different operating conditions. The upper middle plot shows the uncompensated performance levels of each operator. The best value is colored green. The values smaller than the best value obtain colors from green via yellow to red. The worst value obtains the red color. Based on the uncompensated performances, one could think that OP9 has the best performance. However, when accounting for the operating conditions by using (5.7), the order of the performances change (rightmost plot in the upper row). Now the best operator is OP8.

The distribution of the 100-sample average values of the compensated performance x_ζ is shown in the lower left histogram. Clearly, most of the values are between -2 and 2. The GEV probability density distribution fitted to the training data, in addition to the values of the compensated performances for each operator are shown in the lower middle plot. Because the high values of the overall performance V1 are regarded as good, the cumulative distribution (5.10) can be used directly as the evaluation function. The GEV cumulative distribution

Table 5.6: Number of cases H_0 for pairwise Kruskal-Wallis tests where the null hypothesis is rejected in 3 % level of significance. In "Total" line the maximum value is 72.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
OP1	6	7	8	6	7	3	6	4	4	6	6	7
OP2	6	6	6	5	6	4	4	6	8	7	3	7
OP3	-	-	-	-	-	-	-	-	-	-	-	-
OP4	8	6	6	7	6	5	4	4	5	5	5	8
OP5	6	7	6	7	6	4	6	2	7	5	6	8
OP6	4	7	7	7	7	7	5	4	7	7	6	8
OP7	6	7	6	8	5	3	3	2	7	5	6	6
OP8	6	6	8	6	7	3	3	6	4	6	5	6
OP9	6	5	6	6	5	7	6	3	5	3	4	6
OP10	6	7	5	6	3	8	1	1	3	2	5	6
Total	54	58	58	58	52	44	38	32	50	46	46	62
Percentage	75	81	81	81	72	61	53	44	69	64	64	86

function along with the operators' average performances are shown in the rightmost plot on the lower row.

The operating point compensation approach was used for each performance variable prior to the statistical learner model. All variables had at least a small dependency on the operating point, especially with regard to the variance.

5.6.4 Skill index hierarchy: from measurements to skill components

After the operating point compensation, the statistical evaluation of each skill variable for each operator was obtained from the trained GEV distributions. For the skill evaluations, one signifies a good value. The skill evaluations of the operators are shown in Figure 5.11 as a performance matrix. Green values of the cells signify skill variable values close to one. Yellow color is obtained when the variable value is close to 0.5, and the red color is a results of a variable value close to zero.

When looking at the operators with the best performance, that is OP1, OP2 and OP8, one can note that there are clear differences between the green patterns. The operators OP2 and OP8 obtain green values from V8–V12, whereas the operator OP1 obtains red values from V8 (Main task 3 strategy 2), V10 (Main task 4 decision success) and V12 (Main task 3 complexity). The operator OP2 obtains poor values for the most important machine control indicators (V3 and V5), which are green for OP1 and OP8.

The differences between the task level skill component evaluations can more easily be seen from Figure 5.12, where the aggregation is performed using (5.14) and the variable weights described in Table 5.4. The operator OP1 is very good in Main task 1 in machine control as well as planning and decision making. On the contrary, OP8 obtains only mediocre values from planning and decision making with respect to first and second main tasks, whereas OP1 and

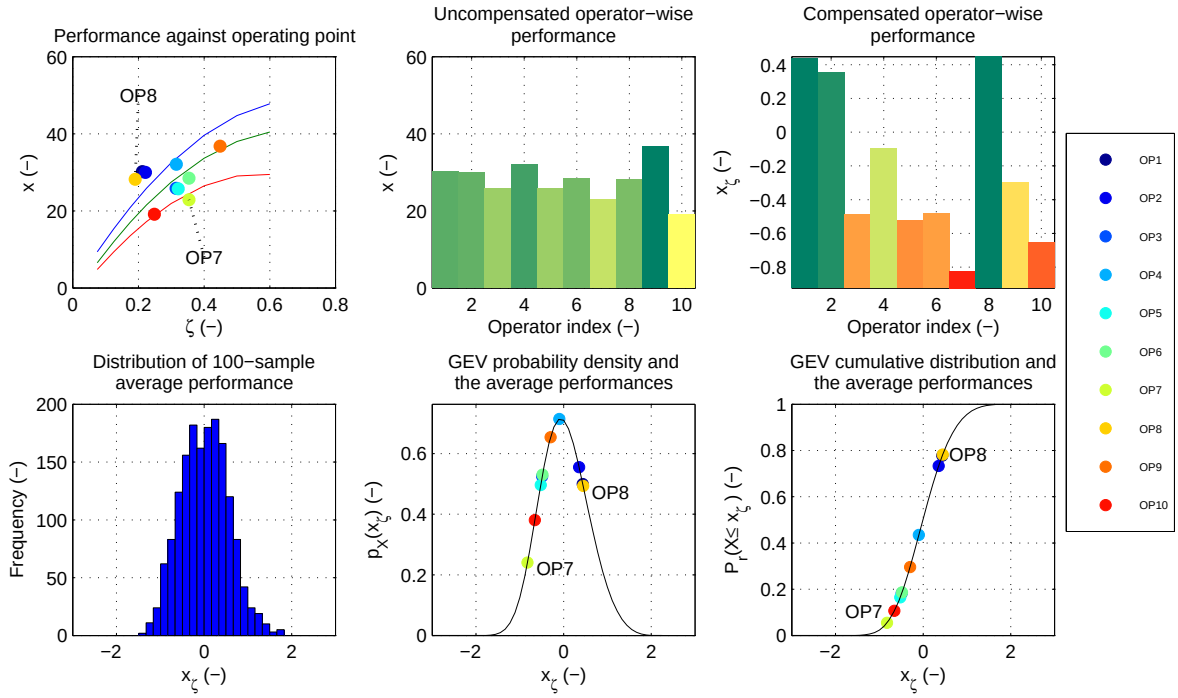


Figure 5.10: Illustration of the statistical learner model for the performance variable V1 (Overall productivity).

OP2 obtain good values from both.

The skill component level evaluations are shown in Figure 5.13, where the aggregation is again performed using (5.14) and the weights shown in Table 5.4. Based on the three highest level indices it is easy to compare the operators. The operator OP1 has very high machine controlling skills, but only mediocre work technique and strategy as well as planning and decision making. Thus, in order to improve the performance of OP1, the planning and decision making as well as the work technique and strategy with respect third and fourth main tasks should be improved. The operator OP2 has very good work technique and strategy as well as planning and decision making skills, but the machine controlling skills are below average. For this operator the most important improvement potential is in the machine controlling skills.

The overall evaluations of OP8 are similar to those of OP1. However, when looking at the main task evaluations, one can see that whereas OP1 has the most significant imporovement potential in third and fourth main tasks, the operator OP8 should imporove first and second main tasks. All in all, it seems that the operators who obtain low evaluations in the overall productivity obtain also mediocre or poor evaluations in all skill components. An exception is OP9 who achieves poor overall productivity but regardless of that, the evaluations of work technique and strategy as well as planning and decision making are good. This is mostly due to first and second main tasks. The most important improvement potential for OP9 are the general machine controlling skills, as well as the work technique, planning and decision making

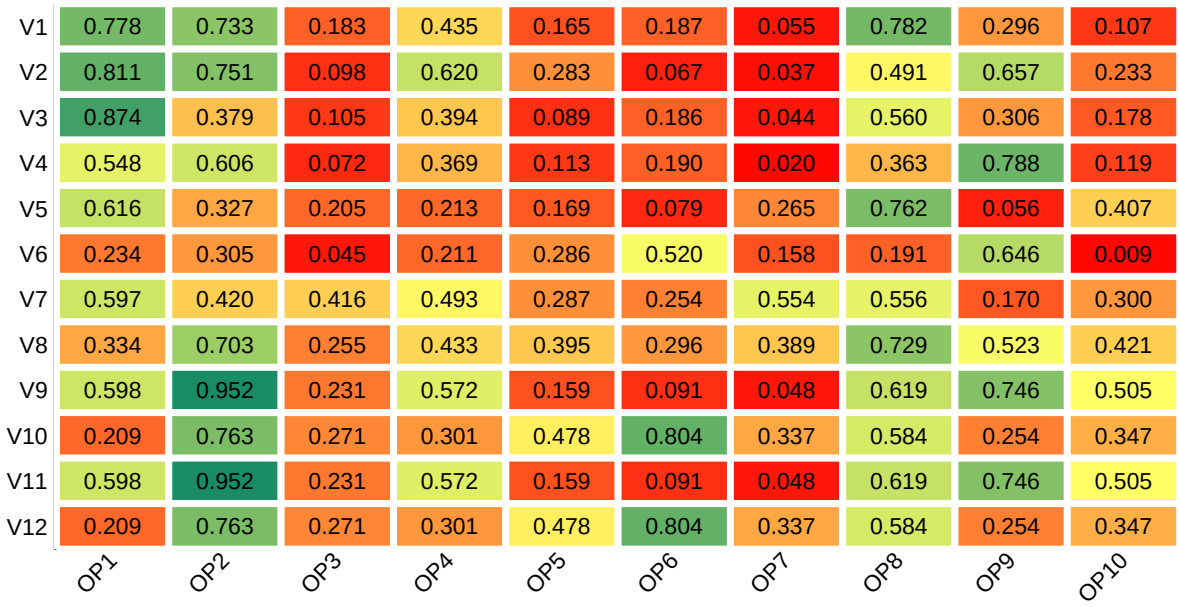


Figure 5.11: The skill evaluation results of variables in Table 5.4 for the operators. One signify a good value for each index.

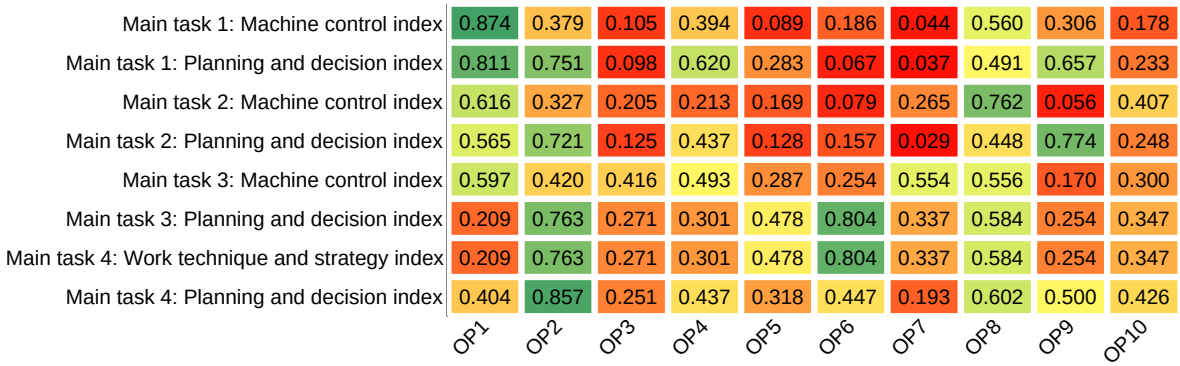


Figure 5.12: The skill component evaluation results of the main tasks for the operators.

Figure 5.13: The overall skill component evaluation results for the operators.

in third and fourth main tasks.

5.6.5 Skill indices and machine performance indices

Although most of the skill indices are designed so that they describe the human performance, not machine performance, the variation of machine's technical performance might have an impact on the maneuverability of the machine. Thus, it is important also to account for the machine's technical condition when analyzing the human performance.

The normalized performance index approach for assessing the technical performance of the forest machine was developed and implemented in [57, 56]. The normalized performance index method was applied to obtain an estimation of the mechanical and hydraulic performance of the machine. The parameters of the indices were obtained based on about 500000 data points for each machine model.

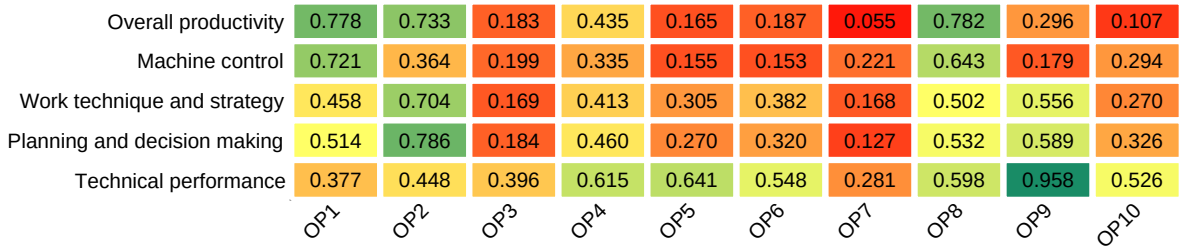


Figure 5.14: The overall productivity, the overall skill component evaluations and the technical performance evaluation for the operators.

The overall productivity, the overall skill component evaluations as well as the technical performance evaluations are shown in Figure 5.14. One can note that OP9 obtains poor skill evaluation results even though the machine's technical performance is remarkably high. Operator OP10 works with the same machine as OP9, but the technical performance of the machine when OP10 was working, is only mediocre. In Table 5.3 it is shown that OP1, OP2 and OP3 work with the same harvester. The technical performance of the harvester seems to be approximately same for all operators. Only the operators OP1 and OP2 achieve good overall productivity. However, it must be remembered that in the data set, OP1 and OP2 had significantly more data than OP3.

In order to illustrate the dependency between the overall productivity and the skill and technical performance evaluations, the operator-wise evaluations are shown in Figure 5.15. In the figures, a circle describes the position of an operator in the space spanned by the performance indices. The greater and the greener the circle is, the better the overall productivity. Based on the results one can note that the technical performance and the skill component evaluations do not seem to correlate significantly.

Because the overall productivity is the most important factor from the perspective of machine owners, a work shift wise productivity regression model was built using the linear regression

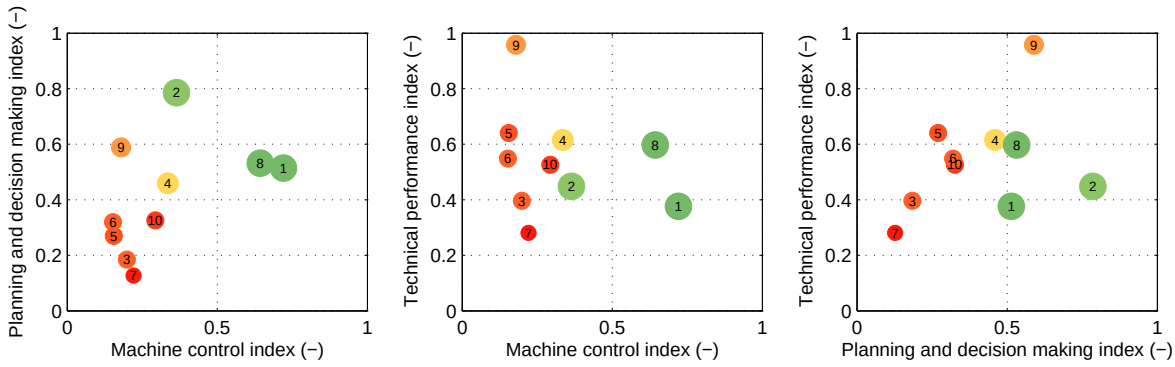


Figure 5.15: The machine control, planning and decision making, and the technical performance indices plotted against each other.

Table 5.7: The coefficients and the corresponding p-values for the productivity regression model.

Variable	Coeff	Std.Err.	Significant	p-value
Machine control	0.6667	0.0895	Yes	$2.5 \cdot 10^{-9}$
Work technique and strategy	-0.2638	0.5212	No	0.6154
Planning and decision making	0.8863	0.1025	Yes	$4.8 \cdot 10^{-11}$
Technical performance	0.0469	0.1040	No	0.6546

method [101]. The objective was to analyze the effect of the skill component level evaluations and the technical performance index to the overall productivity. The analysis was performed as follows. The work shift-wise values of the evaluations for the machine controlling index, the work technique and strategy index, the planning and decision making index, and the technical performance index were combined into a matrix with size 93×4 (in total, there are 93 work shifts of data as shown in Table 5.3). In addition, the work shift wise overall productivity evaluation was combined similarly to a 93×1 matrix. Then, every second of the data point $(1, 3, 5, \ldots, 93)$ were chosen for the training data set and the other half $(2, 4, 6, \ldots, 92)$ for the validation data set. The regression model was estimated by using the training data set and the prediction capability of the model was tested using the validation data set.

The coefficients of the regression model (excluding the constant term) are shown in Table 5.7. Based on the p-values given by the t-test statistics, only the machine control and the planning and decision making had a significant effect on the overall productivity. Thus, only they were included in the final model. An interesting note can be made from the coefficient of the work technique and strategy, which is negative. Because the high values of the evaluations should describe good performance, the negative coefficient for the work technique and strategy could be interpreted that the higher the value the worse the productivity. Thus, the work technique and strategy as such might not be a reliable skill index. However, one should be careful when

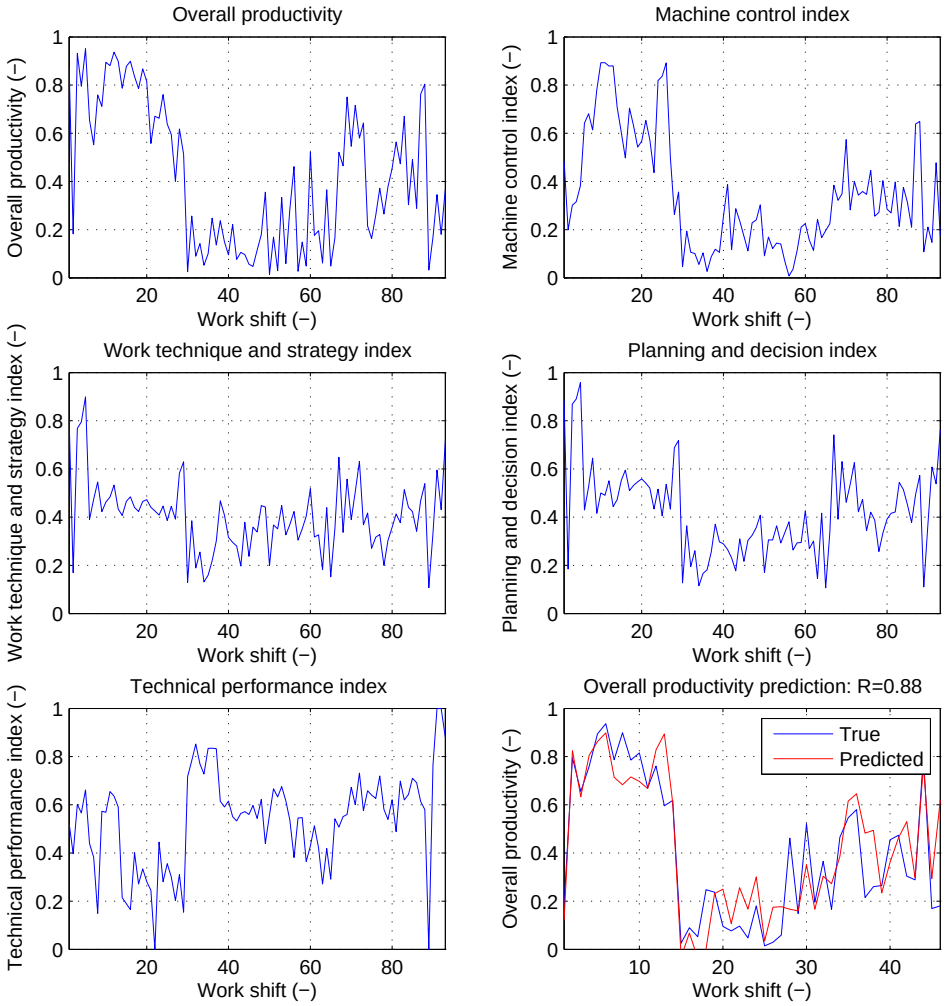


Figure 5.16: The work shift-wise skill evaluation results, technical performance evaluation results, and the productivity prediction accuracy.

interpreting this. As it is explained in Table 5.4, the work technique and strategy evaluation contains two strategy variables from the second main task. In the table, the weights for those variables were set lower than for the other variables, because their reliability was not very high. The problem here is that in the current setting of the statistical learner model, either the low or high values can be set to signify desirable values. However, for the strategy values, it might be that the good value for the strategy variables is closer to the middle. In the future, this should be studied further.

The variables chosen for the final regression model (machine control and planning and decision making) were used to predict the values of the overall productivity for the validation data. The original data set, including the training and the validation data is shown in Figure 5.16. In

Table 5.8: The RTGs and the number of work cycles of different tasks used in the analysis.

Crane	Truck to stack	Stack to truck	Stack to stack	Total
RTG 1.1	1060	5595	4548	11203
RTG 1.2	916	6079	4679	11674
RTG 2.1	43	190	827	1060
RTG 3.1	1407	1305	2044	4756
RTG 4.1	798	8014	11802	20614
RTG 4.2	6255	9450	15167	30872
RTG 4.3	1088	1002	483	2573
RTG 4.4	1780	3956	4031	9767

the bottom right the true overall productivity and the predicted overall productivity for the validation data is shown. Based on the results, one can see that the model is able to predict the most significant changes in the overall productivity. For illustration, the correlation coefficient was also evaluated between the true and predicted overall productivity of the validation data. The correlation 0.88 is statistically significant.

5.7 Case: performance evaluation of RTG cranes

5.7.1 Setting up the experiment

In the container crane cases no information about the work shifts were available. Thus, detailed analysis concerning the skills of individual operators/work shifts cannot be performed. Therefore, the objective is to use the skill and performance evaluation framework for analyzing the container crane performance variations in general.

The work cycle modeling method implemented in Section 4.5 was applied for each work cycle found in the databases of eight RTGs. The data gathering and the implementation of the work cycle analysis as such is explained in [120] in more detail. The result database of the work cycle analysis contains the task types (“truck-to-stack”/”stack-to-truck”/”stack-to-stack”), normal work cycle classification (based on the StateP measure), and the state time distributions. In addition, the moving distances in three directions for each work cycle were calculated.

The RTGs analyzed in this section are shown in Table 5.8. There are eight RTGs in total, of which RTG 1.1 and RTG 1.2 are from the first harbor, RTG 2.1 from the second, RTG 3.1 from the third, and RTG 4.1, RTG 4.2, RTG 4.3 and RTG 4.4 from the fourth. The data from each RTG are from normal operation during one month to two years, depending on the RTG. There is no detailed information about the working environment of the cranes. The number of work cycles (containers) varies by the cranes. The data set of RTG 2.1 contained only about 1000 work cycles, whereas the data set of RTG 4.2 had over 30000 containers. In total, there were roughly 93000 containers in the data set.

In the RTG work it is typical that there are occasionally long idle times, because the objective of the RTG is to support the loading and unloading of the ships performed by the STSs and

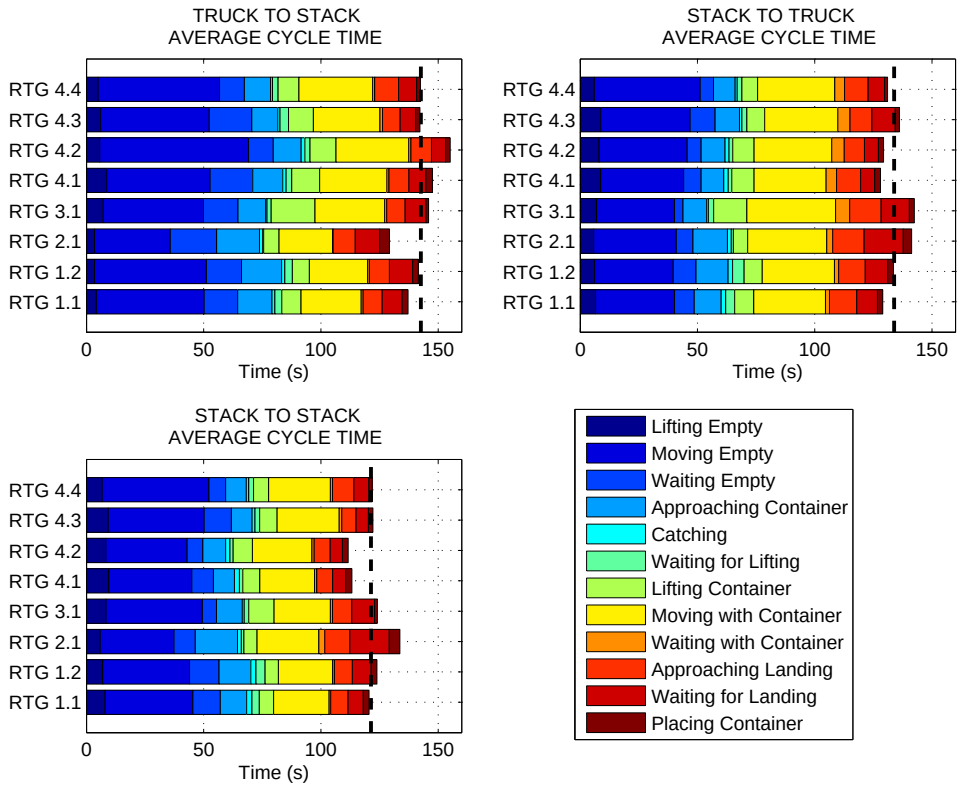


Figure 5.17: Comparison of task time distributions of the RTGs for different task types.

manage the container yard. When there are no other tasks, the RTG can shuffle the containers in the stacks, but typically RTG is not the bottleneck of the logistic chain. Hence the long idle times. Since the objective of this case study is to analyze the performance of the RTGs during normal operation, the work cycles which included long idle times (over two minutes) were omitted from the analysis. In addition, the work cycles with total duration over five minutes were removed from the data.

5.7.2 Comparison of average work cycle time distributions

The comparison of the work cycle time distributions of the RTGs for all task types is shown in Figure 5.17. The average durations of the work cycles over the RTGs are denoted by the thick dashed horizontal lines. The truck to stack is the slowest task type. Based on the time distributions this is due to longer moving-empty times. Thus, the RTG travels longer distances to fetch the container from the truck. The RTG 2.1 has the shortest truck to stack cycle, and the RTG 4.2 the longest. However, in order to determine which RTG is the best, one should account for the moving distances. This is done later in the performance index analysis. However, the work cycle times should be comparable within the harbor.

As for the stack-to-truck task type, the RTG 4.1 obtains the shortest cycle. In general, the average cycle times for this task type do not vary significantly within the harbors. It is interesting that the stack-to-truck task type takes longer for RTG 2.1 than the truck-to-stack task type. In general, the truck-to-stack and stack-to-truck task types differ from each other in waiting times. In the truck-to-stack, there is practically no waiting with container, but more waiting without the container. This is because the RTG typically needs to wait the truck to settle on the truck line so that the container can be picked. If the cooperation between the truck and the RTG is fluent, the container picking operation is fast. Once the container is caught, the RTG typically transfers it directly to the stack, without waiting. In the stack-to-truck task type, there can be more waiting with the container, since the RTG might have to wait for the truck. In addition, the approaching landing (with the container) and waiting for landing times might grow large if the cooperation between the truck and the RTG is not fluent.

The stack-to-stack is the fastest task type, which is typically performed when there are no trucks coming in or out. This is preparation work for loading and unloading of a truck, and it is performed only when needed. Also in this task type, there is typically no waiting with the container. Once the container is caught, it is transferred directly to the target stack. In this task, the RTG 4.2 seems to be the fastest.

5.7.3 Work performance indices for container cranes

In order to analyze the performance variations of the cranes, work performance indices were designed. The indices are time-based and they are computed using the task efficiency approach (5.1). The indices whose values depend on the measurable environmental conditions are compensated using the operating point compensation approach described in Section 5.3. The task times are obtained by the work cycle recognition method implemented for RTG and STS in Sections 4.5 and 4.4.

The indices for the crane performance analysis are shown in Table 5.9. In the table, the first column is the index name, the second presents the definition, the third the aspects of work the index measures, and the fourth the required compensation variables. The overall productivity is the highest level performance indicator. If this index is high, the productivity is at the acceptable level, and there is no need to study the other indices. In the overall productivity index, all factors are reflected, including the machine performance, the operator's performance, and the cooperation of the cranes (the logistic chain performance), as well as the cooperation of the machine operator and the crew (especially in the STS case). The overall productivity index is compensated using the orthogonal sum of the distances between the catching and releasing coordinates as well as the highest point of the container during the work cycle. In the compensation it is also taken into account that the speeds of movements in the gantry, hoist, and trolley directions are different.

The moving speed index measures basically the average speed the as operator transfers the container from the catching point to the releasing point. Although this is affected also by the technical performance and settings of the machine, the effect of human operator for this index is also strong. Some operators might not drive the crane as fast as possible, especially when they know that, for example, the truck is not ready yet. Thus, the poor value for this index might indicate good planning skills of the human operator. However, since there was no information

Table 5.9: Work performance indices for the container cranes.

Index name	**Definition**	**Measures**	**Compensation**
Overall productivity index	The sum of all task times	Machine performance, operator performance, cooperation of the cranes, cooperation between the human operator and the crew	The total (orthogonal) movement distance based on the catching and releasing positions (in the cartesian space) and the highest point during the cycle.
Moving speed index	The sum of Moving empty and Moving with container times	Machine performance and operator performance	Movement distance in the cartesian space (orthogonal)
Container catching index	The sum of Approaching container, catching, and waiting for lifting times	Operator performance, cooperation of the cranes, cooperation between the human operator and the crew	Approaching distance
Container releasing index	The sum of Approaching landing, releasing, and waiting for releasing times	Operator performance, cooperation of the cranes, cooperation between the human operator and the crew	Approaching distance
Waiting with container index	Waiting with container time	Operator performance, work efficiency, cooperation of the cranes, cooperation between the human operator and the crew	None
Waiting without container index	Waiting without container time	Operator performance, work efficiency, cooperation of the cranes, cooperation between the human operator and the crew	None

about the work shifts in the database, this aspect was not studied in this thesis. The moving speed index is compensated by the orthogonal sum of the differences between catching and releasing points.

The container catching index is an important indicator of the human operator's machine controlling skills. Based on the harbor visits and interviews, as well as the video matrial, it is clear that some operators can handle the crane better than the others. This can be seen especially in tasks which require special care, such as catching and releasing tasks. In addition to the human performance, this index measures the efficiency of the cooperation between the cranes, as well as the land and ship staff assisting the crane (especially in the STS case). This index is compensated by the approaching length, that is, the vertical distance between the beginning of the approaching and the container catching point.

The container releasing index is similar to the catching index. It is also a very important human skill indicator. Depending on the task type and the crane, this can measure also the efficiency of the cooperation between the cranes, as well as the crane operator and the assisting staff. Also this index is compensated by the approaching length.

The waiting indices (waiting with container and waiting without container) describe the efficiency of the work. Depending on the current task type, and the crane, the occurrence of waiting can be because of the human operator, work management, poor cooperation between the cranes, or poor cooperation between the assisting staff and the crane operator. In addition, several other types of confusions and exceptional situations might cause waiting. The waiting indices do not need any compensation.

After the operating point compensation, the statistical learner model method is applied to obtain the statistical performance level for the current sample set. The size of the sample set could be, for example, one work shift but in this case the information was not available. Instead, the window size of 30 containers was used. Thus, for each set of 30 containers, the average value is calculated and the statistical performance level is obtained from the GEV cumulative distribution function. In addition, for visualization purposes one can use piecewise smooth scaling to suppress the variations of the high index values as shown in [56] in the normalized performance index case.

5.7.4 Example time series of the RTG work performance indices

The work performance indices were implemented on top of the work cycle modeling system. The parameters of the operating point compensation and the statistical learner model were trained using all data from all RTGs shown in Table 5.8. The window size for the statistical learner model was set to 30 containers. The parameters of the distributions were fitted separately for each task type.

As an example, the time series for the work performance indices were computed for RTG 4.4. The truck to stack task type is shown in Figure 5.18, the stack to truck in Figure 5.19, and the stack to stack in Figure 5.20. For each index, one represents the best possible performance according to the cumulative distribution function, that is, all samples in the training data were worse than the current value. On the contrary, zero value is obtained when the evaluation variable is worse than any of the samples in the training data. The indices are color-coded so that green represents a value close to one, and the red close to zero. For visualization, a

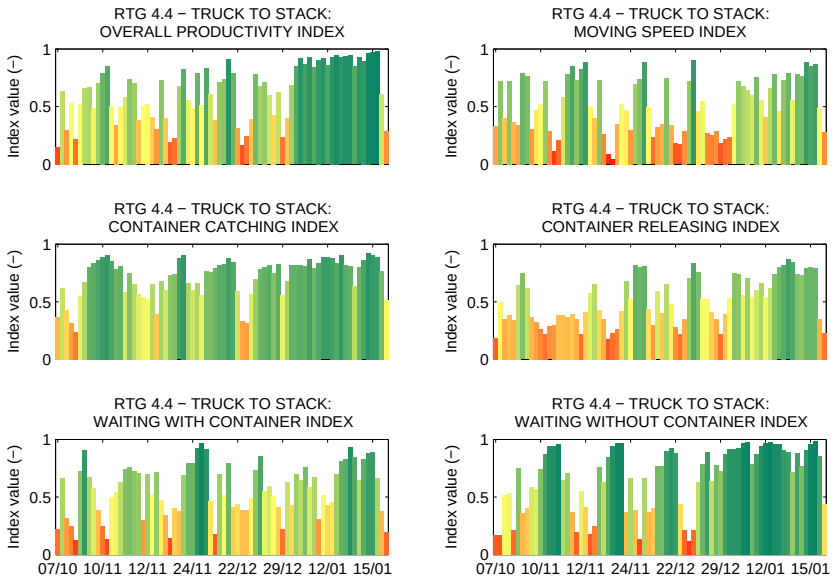


Figure 5.18: Time series representation of the work performance indices for RTG 4.4 truck to stack task type.

piecewise smooth scaling was used to suppress the variation of high index values similarly as it was done in [57]. As a result, an unscaled index value 0.5 is mapped to 0.8, and the values less than 0.5 decrease faster to zero. This was chosen because the value 0.5 represents an average performance with respect to the statistical learner model parameters, which is quite acceptable. Hence, it was mapped to 0.8.

When looking at the index time series in Figure 5.18, one can see that the overall productivity varies significantly. Until the beginning of January, the productivity is only occasionally at a high level. A periodical phenomenon can be seen especially in waiting without container index, where the index is at high level for about six index bars at a time. This corresponds to 180 work cycles, which can be achieved in a work shift. However, when looking at the moving speed index, one can note that it is mostly at higher level when the waiting without container index is low. This can be interpreted that the operators of different work shifts have different working styles. The operators who obtain higher moving speed index tend to wait still until they know that the truck is ready and the next container can be fetched. Once the truck is ready, they move there at full speed. On the contrary, the operators with lower moving speed index do not necessarily wait for the container but they just drive at low speed, because they know that the truck is not ready yet.

Similarly, there are periods where the catching and releasing indices are at low level. By studying the time series, one can see that the overall productivity is rarerly high when the catching or releasing indices are low. The same phenomena can be observed in Figure 5.20 and Figure 5.20.

In conclusion, the indices provide valuable information about the temporal variations of the

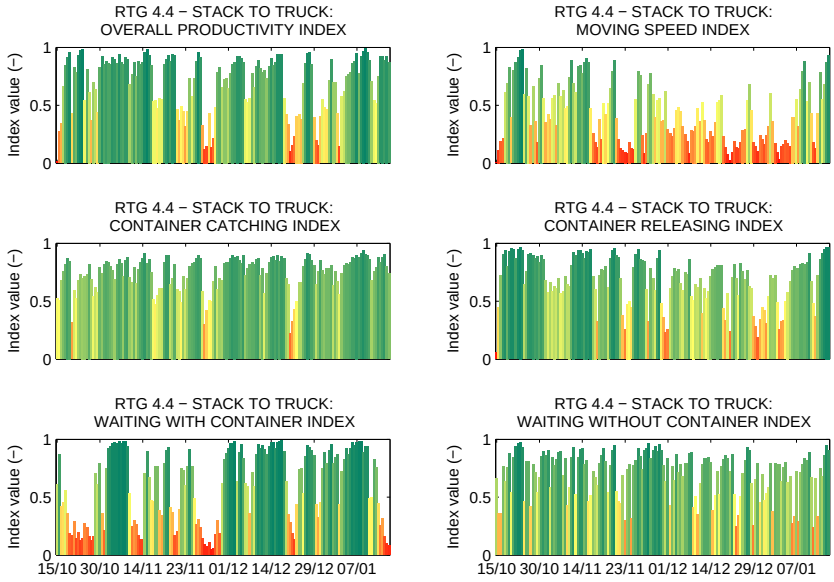


Figure 5.19: Time series representation of the work performance indices for RTG 4.4 stack to truck task type.

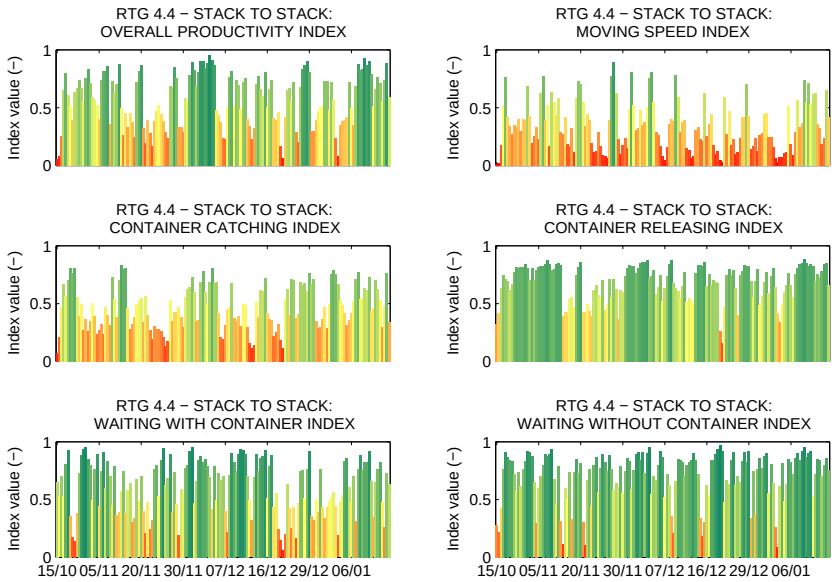


Figure 5.20: Time series representation of the work performance indices for RTG 4.4 stack to stack task type.

Table 5.10: The STSs and the number of work cycles of different tasks used in the analysis.

Crane	Unloading	Loading	Total
STS 1.1	4989	4778	9767
STS 2.1	5193	5655	10848

productivity and the performance of the container transfer process by RTGs. The causes for poor index values can be manyfold, but when used online, the situation in the field is known. Thus, the interpretation of the index values becomes significantly easier.

5.8 Case: performance evaluation of STS cranes

5.8.1 Setting up the experiment

The work cycle modeling method implemented in Section 4.4 was applied for each work cycle found in the databases two STSs. The data gathering and the implementation of the work cycle analysis as such is explained in [120] in more detail. The resulting database of the work cycle analysis contains the task types ("loading"/"unloading"), normal work cycle classification (based on the StateP measure), and the state time distributions. In addition, the moving distances in three directions for each work cycle were calculated.

The STSs analyzed in this section are shown in Table 5.10. There are two STSs in total, from separate harbors. The data from the cranes are from normal operation during two months for both cranes. There is no detailed information about the working environment of the cranes, however it is known that in the first harbor the containers are brought for the STS using straddle carriers (SCs), whereas in the second harbor the containers are brought using road trucks (RTs).

When SCs are used the STS picks the container from the ground, whereas in the RT case the container is typically caught from the platform of the RT. Thus, the container catching (from the platform) is more difficult than from the ground, and the success depends not only on the skills of the operator but also on the positioning of the truck. In the SC case it is also possible that the SC sets the container poorly on the ground, which makes the container catching more difficult for the STS operator. However, typically the container places are marked on the dock, and thus the STS operator is easy to position the crane correctly, provided that the SC operator sets the container correctly on the dock. Similarly as with the RTG case, the work cycles with too long idle times (over three minutes) or with too long total duration (over six minutes) were removed from the data.

In addition, loading on the deck and unloading from the deck is different than to/from the hold, because when operating on/from the deck, additional locks are installed on each corner of the container. In this thesis, only containers operated on/to the hold are considered. The classification for hold/deck is done based on the depth the container is located at the ship side.

5.8.2 Comparison of average work cycle time distributions

The average task time distributions of the STSs are shown in Figure 5.21. In unloading, STS 1.1 is about 40 s faster than STS 2.1. The difference emerges when the STS approaches the container on the ship side. Another significant difference emerges when the STS needs to wait with the container, as well as when the container approaches the platform. Altough there are differences in the working environments, the 40 s difference in inexcusable, resulting in almost 7 containers per hour worse productivity. For unloading a ship of 300 containers, the STS 1.1 performs the work over three hours faster. Here it should be remembered that these are average numbers. The container handling process is a complicated logistic chain. However, the basic unloading work of STS 2.1 is significantly slower than STS 1.1.

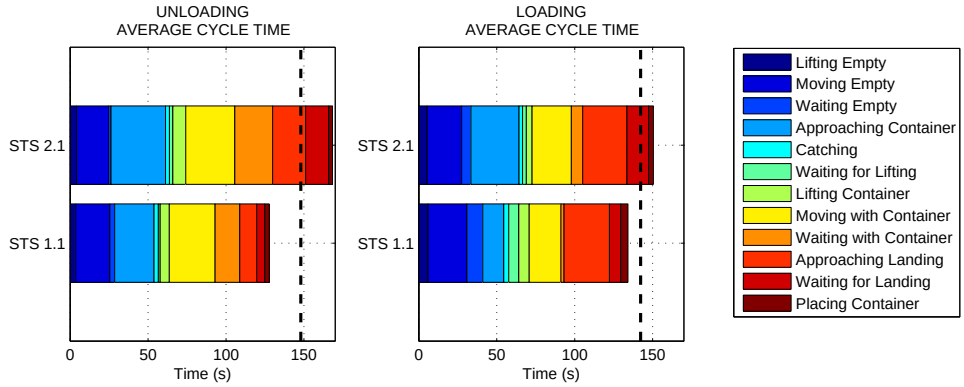


Figure 5.21: Comparison of task time distributions of the STSs for different task types.

In the loading process, STS 2.1 is still worse, but the difference is not that significant. The STS 1.1 is roughly 16 s faster than STS 2.1. Also here, the greatest difference emerges in the approaching task. More specifically, the tasks in which the STS approaches the container on the deck (or on the truck) with the spreader. In addition, waiting for landing is, for some reason, longer for STS 2.1 than for STS 1.1. The differences between the cycle times can be caused by working methods, working instructions, human operators' skills, as well as the cooperation between the crane and the staff. The discovery of the exact cause is difficult without knowing the circumstances at the harbor, but based on the data, it can be concluded that the greatest bottleneck of STS 2.1 in terms of the productivity of the STS is in approaching, catching and releasing of the containers especially at the dock side.

5.8.3 Example time series of the STS work performance indices

Similarly as for the RTGs, the work performance indices were implemented on top of the work cycle modeling system. The parameters of the operating point compensation and the statistical learner model were trained using all the data from all STSs shown in Table 5.10. The window size for the statistical learner model was set to 30 containers. The parameters of the distributions were fitted separately for each task type.

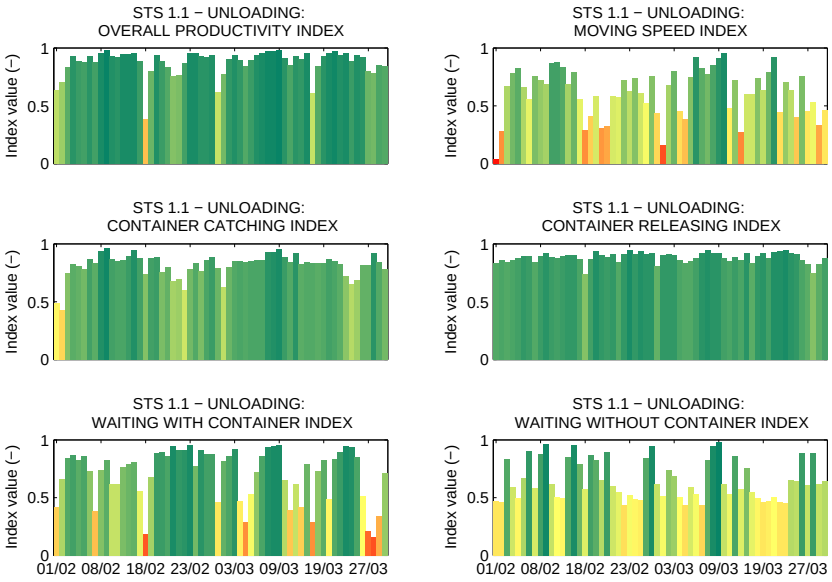


Figure 5.22: Time series representation of the work performance indices for STS 1.1 unloading task type.

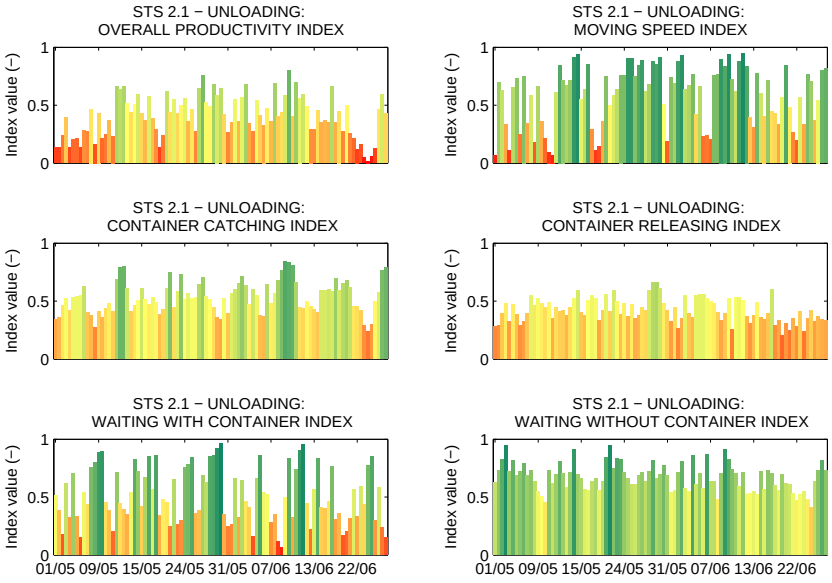


Figure 5.23: Time series representation of the work performance indices for STS 2.1 unloading task type.

The time series of the unloading task type for STS 1.1 is shown in Figure 5.22, and for STS 2.1 in Figure 5.23. As the average task time distributions suggested, the overall productivity of STS 1.1 is significantly better than that of STS 2.1. Comparison of the time series of the indices reveal that the greatest difference between the cranes is in catching and releasing indices. Both operations are significantly slower for STS 2.1 than for STS 1.1. Moreover, when the catching index for STS 2.1 is at green, the productivity is also at green level.

The interpretation of the performance indices with respect to the efficiency of the logistic chain is easier in the STS case than in the RTG case. The objective of the STS is to load and unload the containers as fast as possible. The work of all other vehicles in the harbor should support loading and unloading ships, that is, the STS work. Thus, the indices can be used indirectly to reason the efficiency of the logistic chain and the cooperation of the cranes and the staff. For example, when unloading the ship, the STS normally waits with the container only if the SC has not fetched the previous container yet, or in the RT case, the truck is not ready to receive a container. Although the performance of STS 1.1 is constantly relatively high, there are periods when the waiting with container index is low, meaning that the output flow is not fast enough. Thus, the STS needs to wait, and the logistic chain becomes inefficient. Similarly, one can use also the moving-speed index and waiting-with-container index to determine the efficiency of the logistic chain at the ship and the dock side.

In the STS case, only two cranes from separate harbors were available. The parameters of the performance evaluation method were trained based on the same data. Based on the data used in this case, it is difficult to draw too strong conclusions about the bottlenecks or the improvement potential of the cranes. However, the temporal performance variations and the possible causes within one harbor can be analyzed easily using the work performance indices.

5.9 Discussion

Since the skill evaluation is an essential step in the development of HAMC, the skill and performance evaluation framework for human operated machines was presented. The method exploits the work cycle modeling method presented in Chapter 4. For development of human skill and performance indices based on the task sequence data, four approaches were given to serve as a starting point. Since the performance of a human operated machine is typically strongly dependent on the operating point (external factors such as moving distances, physical dimensions of manipulated objects, etc.), the operating point compensation method was described. Based on the author's experience in several case studies, it is extremely important to account for the favorability of the operating conditions when evaluating human performance.

Because the human operation contains several sources of uncertainties, which cannot be measured, the concept of statistical learner model is used to determine the most likely performance level of the operator. Moreover, because it is impossible to define direct measures for certain skill components or areas of human performance (machine controlling skills, etc.), the evaluation system should be designed so that the higher level indices describing different skill components are combined from lower level indices measuring indirectly those components at the task execution level.

The developed skill evaluation framework was implemented succesfully in three industrial applications. In each case, only the physical measurements readily installed in the machines were

used. Thus, the developed methods can be easily implemented to run online during normal work of the machines.

In the forest harvester case, it was shown that the differences between the skill index levels of different operators are statistically significant, although all variation caused by the external conditions cannot be compensated with the current measurement capability. It was also determined that the variation in the technical performance of the machines could not explain the operator-wise differences between the skill indices. Expert operators achieved high performance regardless of the machine condition.[15] The regression model built to predict the work shift-wise productivity of the harvester based on the skill and machine performance indices achieved significantly high coefficient of determination. The skill evaluation concept developed for forest machines can be used to realize the *performance evaluation* and *skill evaluation interaction* levels for HAMC. In addition, the skill evaluation can be used as a basis for the *coaching* (as described in [119]) and *adaptation* interaction levels (described later in Chapter 8).

In the container crane cases, the information about the work shifts was not available. However, the skill evaluation approach could still be implemented based on windowed data, where one sample was comprised of 30 successive work cycles (regardless of possible shift changes within the sample). For evaluation of the crane (operator and machine) performance in real-time, six performance indices were designed. The time series of the indices revealed that the temporal variation of the performance is significant, and at least part of the variation is possibly caused by the skill (and other human factors such as motivation) differences between the operators and the staff in different work shifts. In this thesis, the sophisticated skill component-wise evaluations were not developed for the container crane cases, because without work shift information they could not be validated. However, already the indices developed in this thesis can reveal the possible causes of bottlenecks (crane operator, input/output flow, assisting staff, etc.) in the crane work. The indices developed for the container cranes realize the *performance evaluation* interaction level of HAMC. Moreover, as the overhead crane example in Section 5.1.2 illustrated, the development of more refined skill indices for the crane operation can be done similarly using the framework presented in this chapter.

When developing a human performance evaluation system to be used in real-time in normal working environment, it is extremely important that one is familiar with the work domain and the operator's working environment. In order to be practically applicable, the human operators of the machines should understand the meaning of different indices and metrics. Moreover, when interviewing the human operators during their normal work, one finds out that there can be natural explanations for some strange phenomena seen in the data. If the skill evaluation system is not designed to account for such exceptional situations, the human operators would not trust to the results, and the system would become useless.

[15] To the author's knowledge, the machines were not broken. If the machine is in such bad condition that the work cannot be performed, naturally even experts cannot achieve high performance.

6 Structural methods for human operator modeling

6.1 Background and objectives

For development of the active and passive adaptation interaction levels of HAMC, the human operator modeling is an important challenge to be solved. This chapter studies the human operator modeling problem from the perspective of structural models. That is, models whose structure is based on the model of the controlled machine. Before going deeper in the modeling methods, let us consider a simple example, which illustrates the desirable properties of the structural models from the machine work perspective.

In the machine work, the operating points of the system vary. For example, if the mass of the load a crane transfers changes, the human operator can take it into account when controlling the machine. Based on experience, a human operator knows that the change in the load mass or the length of the rope changes the dynamics of the crane system. Consequently, the human operator is able to change his/her mental model of the system. For modeling a human operator in machine work one should have a model which can take into account the change in the dynamics of the controlled machine. In other words, the model should be able to predict the human behavior in several operating points, not only in the conditions the model was trained. The optimal control models have this property. They retain the mapping from the parameters of the controlled machine to the human operator model via the Riccati equations [32]. Thus, they can be used to describe a continuum of human operator models with respect to the operating point.

As an example, let us consider the simple trolley crane described in Section 7.1. Consider three cases: baseline case, double mass case, and longer rope case. In the baseline case, the mass of the load is 2.0 kg and the rope length is 1.0 m. The other parameters are as described in Section 7.1. In the double mass case the rope length is 1.0 m but the mass of the load is 4.0 kg. In the longer rope case the mass of the load is 2.0 kg but the rope length is 1.4 m. Setting the Modified Optimal Control Model (MOCM) parameters as described in Appendix B.3, the human operator models for each operating point can be generated. The Bode diagrams of the resulting model adapted to each operating point are shown in Figure 6.1. In each model, the human operator parameters are equal, only the operating point has been changed (and the corresponding gain matrices have been resolved). It is easy to note that the changes in the operating point cause significant changes in the human operator model.

The effects of the operating point changes in the time domain are shown in Figure 6.2. In the top row, the baseline model is used to simulate the unit step response of the trolley crane system with the baseline parameters. Thus, the internal model of the human operator matches with the model of the controlled machine. Even though the MOCM parameters of this example are not identified using real data, one can see that the response of the human operator looks rather realistic. In the middle row, the same baseline human operator model is used to simulate the trolley crane system where the load mass is doubled. The green response in the middle row describes the human operator model adapted to the changed operating point (doubled mass). In the bottom row, the baseline human operator model is used to simulate the trolley crane system where the rope length is 1.4 m instead of 1.0 m, where the baseline model is adapted. The green respose describes the model adapted to the operating point where the rope length is

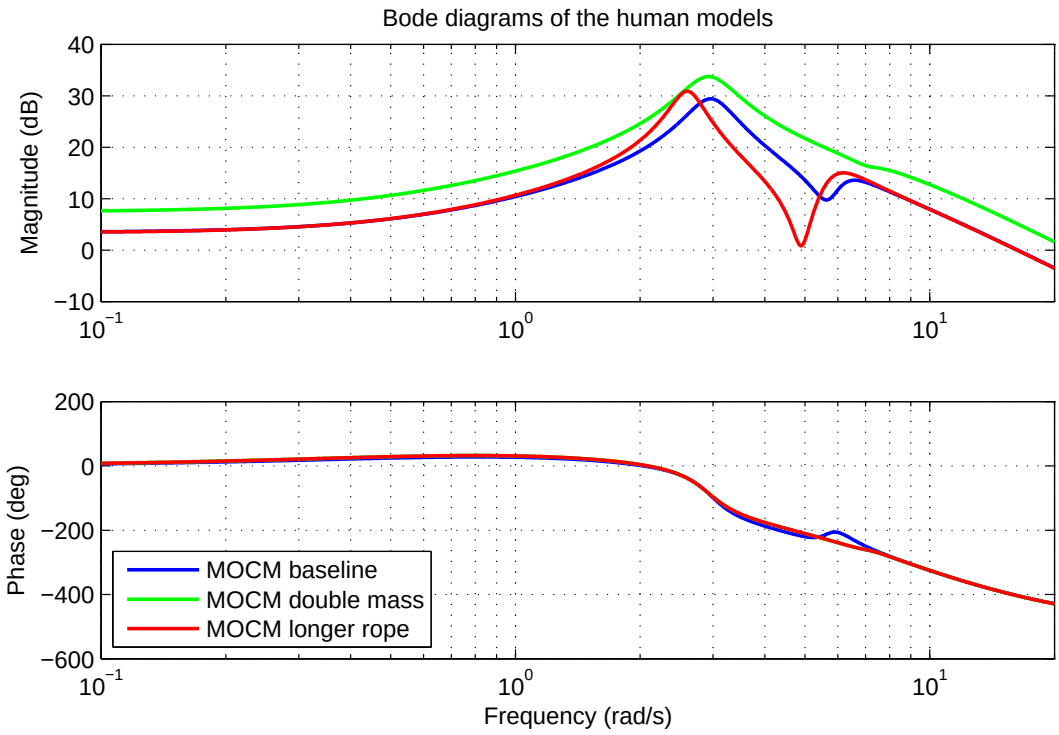


Figure 6.1: Example human operator model in three operating points.

1.4 m. Based on the simulations, it clear that the human operator models used in the machine work should be able to take into account the changes in the machine dynamics.

The optimal control models, such as MOCM have proven to model the human operator well in various flight control tasks [32]. However, in the current for the identification of the model is performed so that values for the human operator time delay, neuromotor lag, noise ratios, disturbance intensities and objective function weights are first guessed, after which the MOCM can be solved. If the guessed parameters do not lead to desirable prediction performance they are "adjusted" [32]. This leads to using direct search methods for the unknown parameters. In addition, if the measurement data are incomplete, that is, all measurement the human operator uses for controlling the machine are not available, the covariance matrices cannot be calculated. Moreover, the MOCM as such does not take into account the possible nonlinearities in the HMI.

The objective of this chapter is to modify the current version of the MOCM so that the nonlinearities in the input and the output could be taken into account, and the unknown parameters could be estimated using gradient-based optimization methods based on normal, possibly incomplete task execution data. The nonlinearities are taken into account using the Hammerstein-Wiener structure, because it provides convenient means of modeling known nonlinearities. In crane operation, for example, the saturation and the rate saturation levels of the control signals migh be known. Thus, it is reasonable to use the knowledge of the nonlinearities in the modeling process. The relaxation of the optimality properties lead to the structure

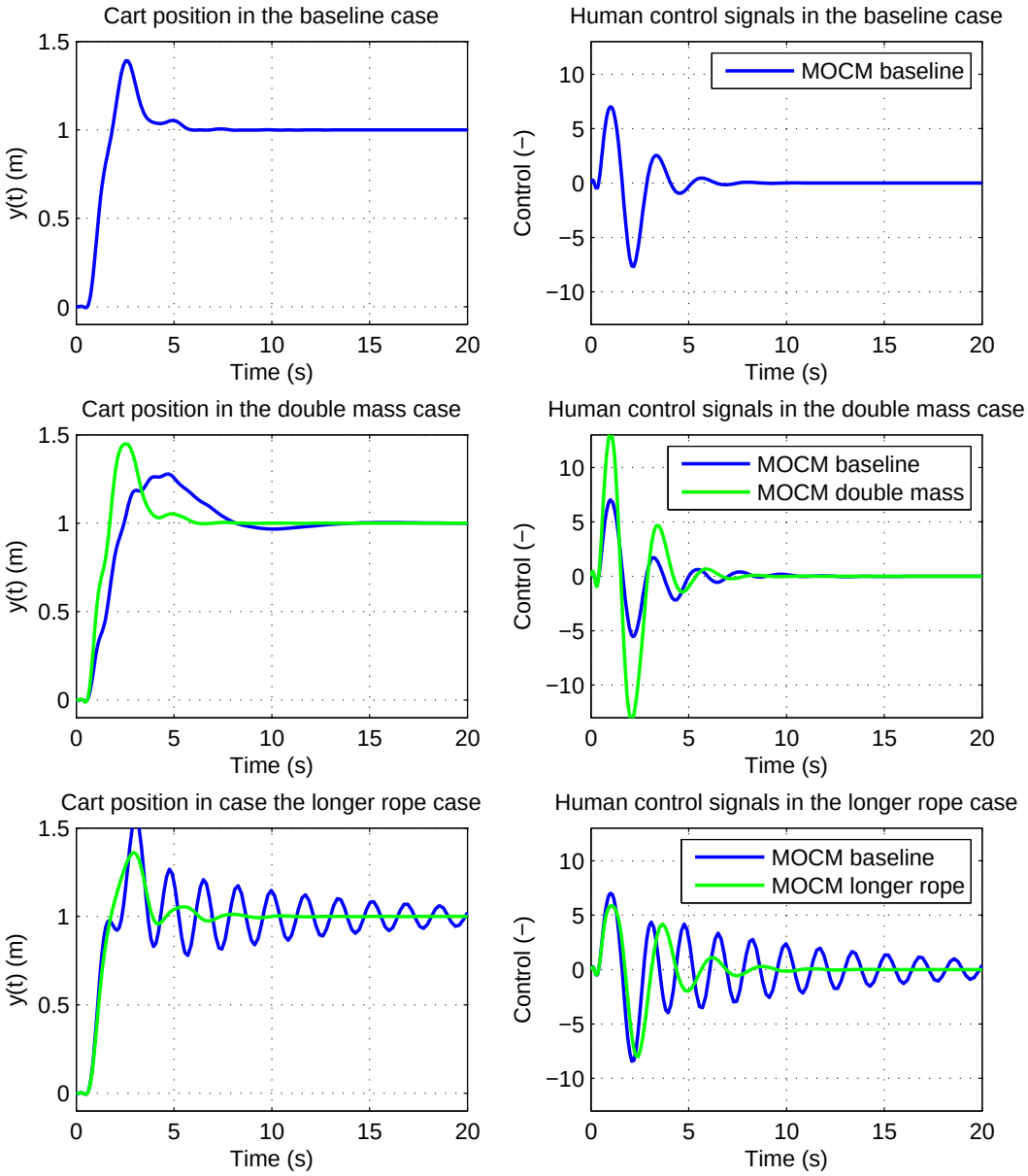


Figure 6.2: The effect of the operating point change in the human controller model in time domain simulation.

of Data-Based State Feedback Model (DBSFM), and gradient-based batch frequency domain identification of the unknown parameters. The frequency domain identification method was used because the derivation of the cost function gradients was easy, and the return difference constraints for preserving the inverse optimality property of DBSFM are expressed in the frequency domain. In order to adapt the model to the changes in the operating point using the

Riccati equations, the weighting matrices for the optimal control, we well as the covariance matrices for the Kalman filter are required. Thus, the Linear Matrix Inequality (LMI)-based inverse optimal control and estimation solutions are presented.

Since the nonlinearities cannot be taken into account in the frequency domain, the discrete version of DBSFM (DDBSFM) is proposed and a gradient-based time-domain identification method is described. For the DDBSFM the inverse optimality criteria could not be found in this thesis. The Hammerstein-Wiener extension for the DDBSFM is described and the corresponding changes in the identification method are described.

The general description of MOCM is given in Appendices B.1 and B.2, which are almost directly cited from [32] and [35]. The MOCM description provides a basis for the new models presented in this Chapter.

6.2 Description of the identification problem and the assumptions

The general identification problem throughout this chapter is to a find mapping from the *available measurements* of the machine $\mathbf{y}(t)$ ($\mathbf{y} \in \mathbb{R}^{N_y}$) to the human operator control signal $\delta(t)$ ($\delta \in \mathbb{R}$) by modifying the unknown parameters θ ($\theta \in \mathbb{R}^{N_\theta}$) (any parameters the model might have). The objective is to model the human operator as a function $\delta(t) = \mathcal{H}(\mathbf{y}(t), \theta)$, where $\mathcal{H}$ is the function of human operator with respect to the measurement variables and the unknown parameters.

The MOCM assumes that the human operator controls the machine in near-optimal manner. It assumes that the human operator can be approximated *on average* using a linear model. That is, when the operator performs the same task several times, the average control signal approaches the response given by the MOCM. The linearity of the human operator is, in general, not a very good assumption. However, in small range of movements, in small range of frequencies, small range of operating points, small deviations from the equilibrium point, and well-defined simple control tasks, the *resultant* behavior of a human as a controller can, *on average* be approximated using linear model with the accuracy sufficient anough for control design [145]. Moreover, if the human operator is assumed to act in near-optimal manner, one could assume that for nearly-linear systems the human is a nearly-linear controller, because an optimal controller for a linear system is linear (this follows from the theory of optimal control). The same assumptions apply for all models studied in this chapter.

6.3 Hammerstein-Wiener formulation of MOCM

In the following, a new formulation of Hammerstein-Wiener MOCM (HWMOCM) is presented based on the MOCM described in Appendix B.1. Traditionally, the nonlinearity in the human operation has been tried to account for by using neural networks, fuzzy inference systems etc. However, as it was discussed earlier, the adaptability to the operating point motivates to use the MOCM-type models in working machine applications. Thus, instead of using black-box models, the MOCM is revised to take into account the effects of possibly known nonlinearities. Even after the proposed modification, the MOCM will not provide accurate results for highly nonlinear systems, because it uses the linearization of the system.

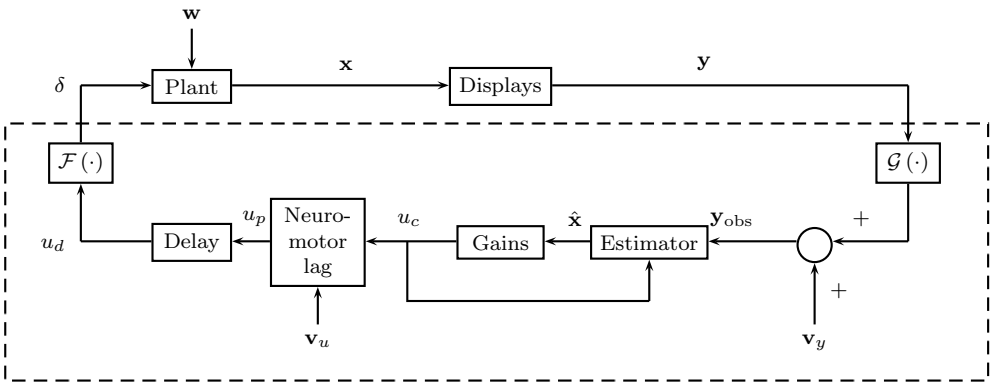


Figure 6.3: A block diagram of the Hammerstein-Wiener MOCM (HWMOCM). The input nonlinearity is described by $\mathcal{G}(\cdot)$ and the output nonlinearity by $\mathcal{F}(\cdot)$.

Sometimes, however, the system the human operator controls might be possible to approximate with sufficient accuracy by using linearization. In such cases, when modeling a human operator controlling an industrial machine such as a crane, the challenge may not be the nonlinearity of the machine but the nonlinearity present in the man-machine interface. That is, the filters and the actuators the control signal passes through from the control lever movements to the movement of the controlled element. In such systems, the core of the human operator model might well be linear but the interface and the input could be nonlinear. These properties can be achieved when using the Hammerstein-Wiener structure. In the Hammerstein-Wiener model the core model is linear but the input signal is transferred through a nonlinear mapping before being directed to the linear dynamical system. The output of the linear system is led through another nonlinear mapping, which produces the output of the Hammerstein-Wiener model. The general represention of the HWMOCM is shown in Figure 6.3. The input nonlinearity is described by a static mapping $\mathcal{G}$ and the output nonlinearity by a static mapping $\mathcal{F}$.

Assume that the MOCM human operator model (B.20) acts as the linear core. The HWMOCM can now be defined by introducing the input and output nonlinearity $\mathcal{G}$ and $\mathcal{F}$ as follows

$$\begin{cases} \dot{\mathbf{x}}_p(t) = \mathbf{A}_p\mathbf{x}_p(t) + \mathbf{B}_p\mathcal{G}\left(\mathbf{y}(t)\right) + \mathbf{E}_p\mathbf{v}_p(t) \\ \delta(t) = \mathcal{F}\left(\mathbf{C}_p\mathbf{x}_p(t)\right) \end{cases} \tag{6.1}$$

with the initial values as $\mathbf{x}_p^T(0) = \begin{bmatrix} \hat{\boldsymbol{\chi}}_0^T & u_{p,0} & \mathbf{x}_{d,0}^T \end{bmatrix}$. The nonlinearities $\mathcal{G}$ and $\mathcal{F}$ are parametrized by $\theta_{\mathcal{G},l}$ $(1 \leq l \leq N_{\mathcal{G}})$ and $\theta_{\mathcal{F},l}$ $(1 \leq l \leq N_{\mathcal{F}})$, respectively. The number of free parameters in $\mathcal{G}$ and $\mathcal{F}$ are denoted by $N_{\mathcal{G}}$ and $N_{\mathcal{F}}$. The discrete-time version of the HWMOCM can be defined as

$$\begin{cases} \mathbf{x}_p(k+1) = \boldsymbol{\Phi}_p\mathbf{x}_p(k) + \boldsymbol{\Gamma}_p\mathcal{G}\left(\mathbf{y}(k)\right) + \boldsymbol{\Gamma}_v\mathbf{v}_p(k) \\ \delta(k) = \mathcal{F}\left(\mathbf{H}_p\mathbf{x}_p(k)\right) \end{cases}, \tag{6.2}$$

where $\boldsymbol{\Phi}_p$, $\boldsymbol{\Gamma}_p$, $\boldsymbol{\Gamma}_v$, and $\mathbf{H}_p$ are obtained by applying the ZOH discretization method for the matrices $\mathbf{A}_p$, $\mathbf{B}_p$, $\mathbf{E}_p$ and $\mathbf{C}_p$, respectively, and the initial values are $\mathbf{x}_p^T(0) = \begin{bmatrix} \hat{\boldsymbol{\chi}}_0^T & u_{p,0} & \mathbf{x}_{d,0}^T \end{bmatrix}$.

6.3.1 Known output level saturation

The human controlled systems are often tuned so that the output level saturates in normal use when the operator is performing large movements. Thus, it is important to account for the saturation in the identification process. Consider a symmetric ideal saturation function SAT

$$\mathrm{SAT}\left(x, x_{\max}\right) = \begin{cases} x, & \text{if } |x| \leq x_{\max} \\ \mathrm{sgn}(x) x_{\max}, & \text{if } |x| > x_{\max} \end{cases}, \tag{6.3}$$

where $x_{\max}$ is the maximum allowable absolute value of x. By using the saturation function, the MOCM including the output level saturation can be written as

$$\begin{cases} \dot{\mathbf{x}}_p(t) = \mathbf{A}_p \mathbf{x}_p(t) + \mathbf{B}_p \mathcal{G}\left(\mathbf{y}(t)\right) + \mathbf{E}_p \mathbf{v}_p(t) \\ \delta(t) = \mathrm{SAT}\left(\mathbf{C}_p \mathbf{x}_p(t), \delta_{\max}\right) \end{cases}, \tag{6.4}$$

where $\delta_{\max}$ is the maximum allowable control signal and the initial values are as in (6.1). The discrete-time equivalent MOCM with saturation is

$$\begin{cases} \mathbf{x}_p(k+1) = \mathbf{\Phi}_p \mathbf{x}_p(k) + \mathbf{\Gamma}_p \mathbf{y}(k) + \mathbf{\Gamma}_v \mathbf{v}_p(k) \\ \delta(k) = \mathrm{SAT}\left(\mathbf{H}_p \mathbf{x}_p(k), \delta_{\max}\right) \end{cases}, \tag{6.5}$$

where the initial values are as in (6.2).

6.3.2 Known output level and rate saturation

In addition to the output level saturation, in the human controlled systems, especially in cranes the derivative of the control signal is often contrained. In cranes, this is done to decrease the swinging of the load due to rapid accelerations caused by large jumps in the control signal.

In continuous-time, the output of (B.20) should be written in differential form

$$\delta(t) = \int_0^t \mathbf{C}_p \dot{\mathbf{x}}_p(t') dt'. \tag{6.6}$$

By using a saturation function SAT, the output with rate saturation can be written as

$$\delta(t) = \int_0^t \mathrm{SAT}\left(\mathbf{C}_p \dot{\mathbf{x}}_p(t'), d\delta_{\max}\right) dt', \tag{6.7}$$

where $d\delta_{\max}$ is the maximum absolute value of the derivative of the output signal δ. The HWMOCM which account for both output saturation and output rate saturation can be written by using two ideal saturation functions as follows

$$\delta(t) = \mathrm{SAT}_L \left(\int_0^t \mathrm{SAT}_R \left(\mathbf{C}_p \dot{\mathbf{x}}_p(t'), d\delta_{\max}\right) dt', \delta_{\max} \right), \tag{6.8}$$

where SAT_L accounts for the output level saturation and SAT_R the output rate saturation.

In order to obtain the output level and the rate saturation formulation for the discrete time

HWMOCM, the output of (6.2) should be written in the increment form, that is

$$\delta(k) = \delta(k-1) + \mathcal{F}\left(\mathbf{H}_p\mathbf{x}_p(k) - \delta(k-1)\right). \tag{6.9}$$

Now, the last term inside $\mathcal{F}$ describes the level the control signal changes in a time step from $k-1$ to k. By using the incremental representation and defining $\mathcal{F}$ as a saturation function, the output of HWMOCM with the rate limit becomes

$$\delta(k) = \delta(k-1) + \mathrm{SAT}\left(\mathbf{H}_p\mathbf{x}_p(k) - \delta(k-1), d\delta_{\max}h\right), \tag{6.10}$$

where $d\delta_{\max}h$ is the maximum allowable increment of the control signal during one time step (h is the sampling rate of the discrete time system). If both, the output level and the output rate saturations are included, the output of the HWMOCM becomes

$$\delta(k) = \mathrm{SAT}_L\left(\delta(k-1) + \mathrm{SAT}_R\left(\mathbf{H}_p\mathbf{x}_p(k) - \delta(k-1), d\delta_{\max}h\right), \delta_{\max}\right), \tag{6.11}$$

where SAT_L is the output level saturation function and SAT_R is the output rate saturation function. Similarly, if the order of the level and rate saturation blocks are reversed the output can be written as

$$\delta(k) = \delta(k-1) + \mathrm{SAT}_R\left(\mathrm{SAT}_L\left(\mathbf{H}_p\mathbf{x}_p(k), \delta_{\max}\right) - \delta(k-1), d\delta_{\max}h\right), \tag{6.12}$$

and the whole HWMOCM in discrete time as

$$\begin{cases} \mathbf{x}_p(k+1) = \mathbf{\Phi}_p\mathbf{x}_p(k) + \mathbf{\Gamma}_p\mathbf{y}(k) + \mathbf{\Gamma}_v\mathbf{v}_p(k) \\ \delta(k) = \delta(k-1) + \mathrm{SAT}_R\left(\mathrm{SAT}_L\left(\mathbf{H}_p\mathbf{x}_p(k), \delta_{\max}\right) - \delta(k-1), d\delta_{\max}h\right) \end{cases}. \tag{6.13}$$

6.3.3 HWMOCM parameter estimation from incomplete data

In [32] a conceptual flow diagram for determination of the MOCM parameters was provided. The method is based on psychological assumptions on signal-to-noise ratios, attention, etc. In the method, the desired signal-to-noise ratios are searched using the fixed point interation. However, the measurements of all outputs the human operator is assumed to observe are needed to compute the signal-to-noise ratios. Moreover, the method used in [32] does not allow the nonlinearities present in the HWMOCM.

In this thesis the parameter estimation is approached from the system identification perspective. Particularly, the interest is in the case when the observation data is incomplete. That is, some signals the human operator uses for control, might be immeasurable. A cost function minimizing the prediction error between the measured human operator control signal and the signal given by the model is first defined. The objective of the parameter estimation procedure is to minimize the cost function.

Given the measured human operator control signal δ_m and the control signal predicted by the model δ, the following cost function is defined

$$V(\theta) = \frac{1}{2}\sum_{k=1}^{N}\left(\delta_m(k) - \delta(k)\right)^2, \tag{6.14}$$

where N is the number of samples of δ_m, and θ ($\theta \in \mathbb{R}^{N_\theta}$) is the vector containing the vectorized unknown parameters of the HWMOCM ($\mathbf{V}_y$, V_u, $\mathbf{W}$, τ_n, τ_p, $\mathbf{Q}_0$, and possibly unknown $\theta_\mathcal{G}$ and $\theta_\mathcal{F}$). In addition, the measurements available from the controlled machine are denoted by $\mathbf{y}$. The sampling step size of the signals is denoted by h.

In this thesis the MOCM parameters are estimated by using numerical optimization. Basically any numerical optimization method can be used. The cost function evaluation for HWMOCM along with the inputs, outputs and the needed operations can be described as follows.

HWMOCM Cost function:

Inputs: Unknown parameters $\mathbf{V}_y$, V_u, $\mathbf{W}$, τ_n, τ_p, $\mathbf{Q}_0$, *and input data* $\mathbf{y}$ *and* δ_m. *The system parameters* $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$. *The sampling rate* h *of the measurement data. Unknown or known parameters* $\theta_\mathcal{G}$ *and* $\theta_\mathcal{F}$ *of the input and output nonlinearities.*

Output: The cost $V(\theta)$ *with the current parameters.*

Operations:

1. *Initialization: Form 0- and 1-plants. Using* (B.9), (B.18), *and* (B.40) *solve* $\mathbf{g}_p$ *and* $\mathbf{F}$ *and form* $\mathbf{A}_p$, $\mathbf{B}_p$ *and* $\mathbf{C}_p$.
2. *Discretization: Using ZOH discretization and the sample rate* h, *form the discrete-time system matrices* $\mathbf{\Phi}_p$, $\mathbf{\Gamma}_p$ *and* $\mathbf{H}_p$.
3. *Evaluation: Using the discretized operator model* (6.2), *the measurement data* δ_m *and* $\mathbf{y}$, *and the cost function* (6.14) *evaluate the fitness of the model for the current set of unknown parameters* θ.

In practice, the control weighting matrix $\mathbf{Q}_0$ can be computed similarly as in (B.11). However, the measurement matrix $\mathbf{C}_s$ should be modified to include the variables which are not measurable but which can be assumed to be measurable by the human operator. That is, if $\mathbf{C}_s = \begin{bmatrix}\mathbf{C} & \mathbf{D}\mathbf{C}_d\end{bmatrix}$ are based on the true measurements $\mathbf{y}$ available in the system (B.1), one can define $\mathbf{C}_a$ and $\mathbf{D}_a$ as the assumed measurements the human operator uses. Thus, by defining $\mathbf{C}_{sa} = \begin{bmatrix}\mathbf{C}_a & \mathbf{D}_a\mathbf{C}_d\end{bmatrix}$, the weighting matrix $\mathbf{Q}_0$ becomes

$$\mathbf{Q}_{0a} = \begin{bmatrix}\mathbf{C}_{sa}^T\mathbf{Q}_y\mathbf{C}_{sa} & \mathbf{C}_{sa}^T\mathbf{Q}_y\mathbf{D}_s \\ \mathbf{D}_s^T\mathbf{Q}_y\mathbf{C}_{sa} & \mathbf{D}_s^T\mathbf{Q}_y\mathbf{D}_s + r\end{bmatrix}, \tag{6.15}$$

where $\mathbf{Q}_y$ is chosen based on the expert knowledge with possibly some unknown elements. Moreover, one needs to redefine the performance index the human controller tries to minimize to account for the estimated states as

$$J_p = \lim_{t\to\infty}\frac{1}{t}\int_0^t \left(\boldsymbol{\chi}^T\mathbf{Q}_{0a}\boldsymbol{\chi} + \dot{u}_p^2 f\right), \tag{6.16}$$

where $\mathbf{Q}_{0a} \geq 0$ and $f > 0$. Otherwise the MOCM solution proceeds as described previously.

Remark 1 *Note that the MOCM is an overparametrized model, that is, generally there are more unknowns than states. Thus, in practice, one should fix* $\mathbf{Q}_y$ *and* τ_n *beforehand based on the knowledge of the task. Thereafter, the other parameters can be searched minimizing* (6.14).

6.4 Data-based state feedback model of human operator

6.4.1 General structure

The Data-Based State Feedback Model (DBSFM) structure proposed in this thesis is a concept to estimate the parameters of the optimal control type of human operator models. The model structure for DBSFM studied in this thesis is similar to the MOCM and the Hess-LQG approximation of OCM (see the appendix of [32] for details). However, in DBSFM the optimality assumptions are relaxed in order to enable the use of faster gradient-based identification algorithms. The optimality can be preserved by introducing certain constraints presented later. In this thesis, multiple input single output (MISO) structure is assumed. The derivation of the DBSFM is simular to the MOCM, except that the control and the estimator gains are not necessarily solutions of the corresponding Riccati equations. In addition, the output delay of the model is kept exact, instead of using the Padé approximation.

The model is deduced from (B.20) except that now the delay is shown exactly in δ rather than introducing a Padé approximation induced state $\mathbf{x}_d$. Thus, the human operator model according to the DBSFM studied in this thesis is given by

$$\begin{cases} \begin{bmatrix} \dot{\hat{\boldsymbol{\chi}}} \\ \dot{u}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 - \mathbf{F}\mathbf{C}_1 - \mathbf{B}_0 \begin{bmatrix} \mathbf{g}_p & 0 \end{bmatrix} & 0 \\ - \begin{bmatrix} \mathbf{g}_p & 0 \end{bmatrix} & -\frac{1}{\tau_n} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\chi}} \\ u_p \end{bmatrix} + \begin{bmatrix} \mathbf{F} \\ 0 \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{F} & 0 \\ 0 & \frac{1}{\tau_n} \end{bmatrix} \begin{bmatrix} \mathbf{v}_y \\ v_u \end{bmatrix} \\ \delta = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\chi}}(t-\tau_p) \\ u_p(t-\tau_p) \end{bmatrix} \end{cases}$$

or

$$\begin{cases} \dot{\mathbf{x}}_p = \mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p \mathbf{y} + \mathbf{E}_p \mathbf{v}_p \\ \delta = \mathbf{C}_p \mathbf{x}_p(t-\tau_p) \end{cases}, \tag{6.17}$$

where $\mathbf{F}$ is the state estimator gain, $\mathbf{g}_p$ is the state feedback gain, τ_n the operator's neuromotor lag, and τ_p the operator's time delay, and the initial values are defined as $\mathbf{x}_p^T(0) = \begin{bmatrix} \hat{\boldsymbol{\chi}}_0^T & u_{p,0} \end{bmatrix}$. In the DBSFM concept, the state estimator and feedback gains, neuromotor lag, and the dead time delay can all be unknown.

6.4.2 DBSFM frequency domain parameter estimation algorithm

6.4.2.1 Frequency domain cost function

Assume that the human operator performs a task described by the reference signal $\mathbf{y}_r(t)$ ($\mathbf{y}_r(t) \in \mathbb{R}^{N_y}$). The plant output is denoted by $\mathbf{y}(t)$ ($\mathbf{y}(t) \in \mathbb{R}^{N_y}$). The human operator's measurable control signal is denoted again by $\delta(t)$ ($\delta(t) \in \mathbb{R}$). The Laplace-transformations of the reference signal, plant output and operator's control signal are $\mathbf{Y}_r(s)$, $\mathbf{Y}(s)$, and $\Psi(s)$, assuming zero initial conditions. Define $\mathbf{Y}_e(s) = (\mathbf{Y}_r(s) - \mathbf{Y}(s))$. Thus, the state-space representation (6.17) of the human operator dynamics (excluding the noise model) can be written in Laplace domain as

$$\begin{cases} s\mathbf{X}_p(s) = \mathbf{A}_p \mathbf{X}_p(s) + \mathbf{B}_p \mathbf{Y}_e(s) \\ \mathbf{Y}_e(s) = \mathbf{Y}_r(s) - \mathbf{Y}(s) \\ \Psi(s) = \mathbf{C}_p \mathbf{X}_p(s) e^{-\tau_p s} \end{cases}. \tag{6.18}$$

This can be transformed into a transfer function representation

$$\begin{cases} \Psi(s) = e^{-\tau_p s}\mathbf{C}_p\left(s\mathbf{I} - \mathbf{A}_p\right)^{-1}\mathbf{B}_p\mathbf{Y}_e(s) = \mathbf{G}_p(s)\mathbf{Y}_e(s) \\ \mathbf{Y}_e(s) = \left(\mathbf{Y}_r(s) - \mathbf{Y}(s)\right) \end{cases}, \tag{6.19}$$

where $\mathbf{G}_p(s)$ is the human transfer function. In frequency domain the estimated frequency function of the human operator's control commands becomes

$$\begin{cases} \hat{\Psi}(i\omega,\theta) = e^{-\tau_p i\omega}\mathbf{C}_p\left(i\omega\mathbf{I} - \mathbf{A}_p\right)^{-1}\mathbf{B}_p\mathbf{Y}_e(i\omega,\theta) \\ \mathbf{Y}_e(i\omega,\theta) = \left(\mathbf{Y}_r(i\omega) - \mathbf{Y}(i\omega)\right), \end{cases} \tag{6.20}$$

where θ ($\theta \in \mathbb{R}^{N_\theta}$) represents the unknown parameters of the model, that is, the state estimator gain matrix $\mathbf{F}$, the control gain $\mathbf{g}_p$, the neuromotor lag τ_n, and human operator's time delay τ_p. The number of unknown parameters N_θ is $(N_a+1)N_y + N_a + 2$, where $(N_a+1)N_y$ is the size of $\mathbf{F}$, N_a the size of $\mathbf{g}_p$, and there are two scalars τ_n and τ_p.

The objective of the identification is to find such a transfer function which best maps the control error frequency function $\mathbf{Y}_e(i\omega)$ to the frequency function of the operator's control signal $\Psi(i\omega)$. If the frequency range of interest is divided into N regions (equally or logarithmically spaced), the cost function to be minimized can be written as

$$V(\theta) = \sum_{k=1}^{N} \varepsilon(i\omega_k,\theta)^H \varepsilon(i\omega_k,\theta), \tag{6.21}$$

where $\varepsilon(i\omega,\theta) = \Psi(i\omega) - \hat{\Psi}(i\omega,\theta)$, and c^H the hermitian operation for the complex number c, and ω_k is the frequency sample at k, ($\omega_k \in]0,\omega_{\max}]$). In practice, $\omega_{\max}$ depends on the sampling frequency but in the human operator modeling case it is enough to set $\omega_{\max} \leq 35$ rad/s, because the fastest spinal feedback loops have at least 30 ms delay [76]. The experimental frequency function of the human operator's control signal is denoted by $\Psi(i\omega)$. It can be obtained by using the periodograms or the FFT.

6.4.2.2 Gradient of the cost function with respect to the parameters

Minimization of the cost function (6.21) can be carried out using gradient-based iterative optimization algorithms. Therefore, the gradient with respect to the model parameters needs to be derived. The gradient of the cost function with respect to the l^{th} element of the parameter vector, that is, θ_l is

$$\frac{\partial V(\theta)}{\partial \theta_l} = \sum_{k=1}^{N} \frac{\partial}{\partial \theta_l}\bar{V}(i\omega_k,\theta), \tag{6.22}$$

where

$$\bar{V}(i\omega_k,\theta) = \varepsilon(i\omega_k,\theta)^H \varepsilon(i\omega_k,\theta). \tag{6.23}$$

Because $\varepsilon(i\omega_k,\theta)$ is an analytic function [124], the gradient of $\bar{V}(i\omega_k,\theta)$ with respect to θ_l can be written as

$$\begin{aligned}\frac{\partial}{\partial\theta_l}\bar{V}(i\omega_k,\theta) &= \frac{\partial}{\partial\theta_l}\varepsilon(i\omega_k,\theta)^H\varepsilon(i\omega_k,\theta)\\ &= \frac{\partial\varepsilon(i\omega_k,\theta)^H}{\partial\theta_l}\varepsilon(i\omega_k,\theta)+\varepsilon(i\omega_k,\theta)^H\frac{\partial\varepsilon(i\omega_k,\theta)}{\partial\theta_l}\\ &= 2\mathrm{Re}\left(\varepsilon(i\omega_k,\theta)^H\frac{\partial\varepsilon(i\omega_k,\theta)}{\partial\theta_l}\right)\end{aligned} \tag{6.24}$$

Substituting (6.24) into (6.22), the gradient of $V(\theta)$ becomes

$$\frac{\partial V(\theta)}{\partial\theta_l} = 2\sum_{k=1}^{N}\mathrm{Re}\left(\varepsilon(i\omega_k,\theta)^H\frac{\partial\varepsilon(i\omega_k,\theta)}{\partial\theta_l}\right) \tag{6.25}$$

In order to evaluate the gradient the term $\frac{\partial\varepsilon(i\omega_k,\theta)}{\partial\theta_l}$ needs to be evaluated. The difference between the values of the true frequency function and the predicted frequency function can be written as

$$\varepsilon(i\omega_k,\theta) = \Psi(i\omega_k) - \mathbf{G}_p(i\omega_k,\theta)\mathbf{Y}_e(i\omega_k,\theta), \tag{6.26}$$

where $\mathbf{G}_p$ is the human transfer function. Using (6.19) this can be written as

$$\begin{aligned}\varepsilon_k &= \Psi_k - e^{-\tau_p i\omega_k}\mathbf{C}_p(i\omega_k\mathbf{I}-\mathbf{A}_p)^{-1}\mathbf{B}_p\left(\mathbf{Y}_{r,k}-\mathbf{Y}_k\right)\\ &= \Psi_k - e^{-\tau_p i\omega_k}\mathbf{C}_p(i\omega_k\mathbf{I}-\mathbf{A}_p)^{-1}\mathbf{B}_p\mathbf{Y}_{e,k}\\ &= \Psi_k - \tilde{\mathbf{C}}_k\tilde{\mathbf{A}}_k^{-1}\tilde{\mathbf{B}}_k,\end{aligned} \tag{6.27}$$

where the shorthand notations $\Psi_k = \Psi(i\omega_k,\theta)$, $\varepsilon_k = \varepsilon(i\omega_k,\theta)$, $\mathbf{Y}_{e,k} = \mathbf{Y}_e(i\omega_k,\theta)$, $\mathbf{Y}_{r,k} = \mathbf{Y}_r(i\omega_k)$, and $\mathbf{Y}_k = \mathbf{Y}(i\omega_k,\theta)$ are used, as well as denoting $\tilde{\mathbf{A}}_k = i\omega_k\mathbf{I}-\mathbf{A}_p$, $\tilde{\mathbf{B}}_k = \mathbf{B}_p\mathbf{Y}_{e,k}$ and $\tilde{\mathbf{C}}_k = e^{-\tau_p i\omega_k}\mathbf{C}_p$. The full description of $\tilde{\mathbf{A}}_k$ is given as

$$\begin{aligned}\tilde{\mathbf{A}}_k &= i\omega\mathbf{I}-\mathbf{A}_p, \text{ where}\\ \mathbf{A}_p &= \begin{bmatrix}\mathbf{A}_1-\mathbf{F}\mathbf{C}_1-\mathbf{B}_1\mathbf{l}_1 & 0\\ -\frac{\mathbf{l}_1}{\tau_n} & -\frac{1}{\tau_n}\end{bmatrix}\\ &= \begin{bmatrix}\begin{bmatrix}\mathbf{A} & \mathbf{B}\mathbf{C}_d & \mathbf{B}\\ 0 & \mathbf{A}_d & \mathbf{B}_d\\ 0 & 0 & -\frac{1}{\tau_n}\end{bmatrix} - \mathbf{F}\begin{bmatrix}\mathbf{C}^T\\ (\mathbf{D}\mathbf{C}_d)^T\\ \mathbf{D}^T\end{bmatrix}^T - \begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}\mathbf{g}_p & 0\end{bmatrix} & 0\\ -\begin{bmatrix}\mathbf{g}_p & 0\end{bmatrix} & -\frac{1}{\tau_n}\end{bmatrix}\end{aligned}$$

The expression (6.27) requires that the matrix $\tilde{\mathbf{A}}_k$ is invertible. This holds for any values of the free parameters $\mathbf{F}$, $\mathbf{g}_p$, τ_p, and τ_n, the matrix $\tilde{\mathbf{A}}_k$, which result in a stable $\mathbf{A}_p$ ($\mathbf{A}_p \geq 0$), because only positive frequencies $\omega_k \in]0,\omega_{\max}]$ are used in the parameter estimation. In order for $\tilde{\mathbf{A}}_k$ to be singular, $\det(i\omega\mathbf{I}-\mathbf{A}_p) = 0$ should hold. However, this holds only when one of the eigenvalues of $\mathbf{A}_p$ equals to zero, which can occur only when $\omega_k = 0$. Because $\omega_k = 0$ is omitted in the estimation, $\tilde{\mathbf{A}}_k$ is invertible within the set of frequencies used in the estimation.

Computing the derivative of (6.27) with respect to θ_l, the following matrix formula is needed. Consider any invertible square matrix $\mathbf{A}(x)$ as a function of a scalar x. The derivative of the inverse of $\mathbf{A}$ with respect to x is [121]

$$\frac{\partial \mathbf{A}^{-1}(x)}{\partial x} = -\mathbf{A}^{-1}(x)\frac{\partial \mathbf{A}(x)}{\partial x}\mathbf{A}^{-1}(x). \tag{6.28}$$

By using the chain rule and (6.28), the gradient of (6.27) becomes

$$\begin{aligned}
\frac{\partial \varepsilon_k}{\partial \theta_l} &= \frac{\partial}{\partial \theta_l}\left(\Psi_k - \tilde{\mathbf{C}}_k\tilde{\mathbf{A}}_k^{-1}\tilde{\mathbf{B}}_k\right) \\
&= -\frac{\partial \tilde{\mathbf{C}}_k}{\partial \theta_l}\tilde{\mathbf{A}}_k^{-1}\tilde{\mathbf{B}}_k - \tilde{\mathbf{C}}_k\left(\frac{\partial}{\partial \theta_l}\tilde{\mathbf{A}}_k^{-1}\right)\tilde{\mathbf{B}}_k - \tilde{\mathbf{C}}_k\tilde{\mathbf{A}}_k^{-1}\frac{\partial}{\partial \theta_l}\tilde{\mathbf{B}}_k \\
&= -\frac{\partial \tilde{\mathbf{C}}_k}{\partial \theta_l}\tilde{\mathbf{A}}_k^{-1}\tilde{\mathbf{B}}_k + \tilde{\mathbf{C}}_k\tilde{\mathbf{A}}_k^{-1}\frac{\partial \tilde{\mathbf{A}}_k}{\partial \theta_l}\tilde{\mathbf{A}}_k^{-1}\tilde{\mathbf{B}}_k - \tilde{\mathbf{C}}_k\tilde{\mathbf{A}}_k^{-1}\frac{\partial \tilde{\mathbf{B}}_k}{\partial \theta_l} \\
&= -\frac{\partial \tilde{\mathbf{C}}_k}{\partial \theta_l}\tilde{\mathbf{A}}_k^{-1}\tilde{\mathbf{B}}_k + \tilde{\mathbf{C}}_k\tilde{\mathbf{A}}_k^{-1}\left(\frac{\partial \tilde{\mathbf{A}}_k}{\partial \theta_l}\tilde{\mathbf{A}}_k^{-1}\tilde{\mathbf{B}}_k - \frac{\partial \tilde{\mathbf{B}}_k}{\partial \theta_l}\right)
\end{aligned} \tag{6.29}$$

Remark 2 *Note that $\tilde{\mathbf{A}}_k$, $\tilde{\mathbf{B}}_k$ and $\tilde{\mathbf{C}}_k$ depend on the model parameters and the frequency. The derivatives with respect to each element of the unknown parameters $\mathbf{F}$, $\mathbf{g}_p$, τ_p, and τ_n can be obtained by using the well-known rules for computing derivatives of the matrix functions while applying* (6.29).

6.4.3 Initial values for the gradient-based algorithm

The feasible initial values for the gradient-based optimization can be set by assuming the optimality of the controller and the state estimator. In MOCM the optimal control gain matrix is obtained by solving (B.9), where $\mathbf{K}$ is the solution of the Riccati equation (B.10), with $\mathbf{Q}_0 \geq 0$ and $f > 0$. Moreover, it is required that $\tau_n = \frac{1}{g_{N_a+1}}$. Therefore, one first decides the initial guess for τ_n. Then the solution of the initial value of the feedback gain matrix requires the solution of the optimization problem

$$\min_f \left(g_{N_a+1} - \frac{1}{\tau_n}\right)^2, \tag{6.30}$$

when g_{N_a+1} is as in (B.9) and f as in (B.10). The weighting matrix $\mathbf{Q}_0$ can be chosen based on the prior knowledge of the task the operator is trying to execute.

In MOCM the Kalman filter gain matrix is obtained by solving (B.18), where $\mathbf{\Sigma}_1$ is the positive definite solution of the Riccati equation (B.19), with $\mathbf{W}_1 \geq 0$ and $\mathbf{V}_y > 0$. The initial guess for the state estimator matrix $\mathbf{F}$ can be obtained by solving (B.18). Choosing $\mathbf{W}_1$ and $\mathbf{V}_y$ as identity matrices of appropriate dimensions will give a suitable initial value for $\mathbf{F}$. In practice, the initial values can be obtained using the LMI approach described in B.4.

6.4.4 Guaranteeing the existence of the inverse solution

To retain the mapping from the parameters of the model of the controlled machine to the human operator parameters in DBSFM, the control and estimation gain matrices are needed for the changed system model. These are obtained by solving the corresponding Riccati equations using $\mathbf{Q}_0 \geq 0$, $f > 0$, $\mathbf{W}_1 \geq 0$ and $\mathbf{V}_y > 0$ which, in turn, are the solutions of the corresponding inverse optimal control and estimation problems. The LMI approach for the inverse optimal control problem is described in Appendix C.1. Unfortunately, the parameter estimation of DBSFM does not necessarily converge to the parameters for which the inverse solution exists. One can guarantee the existence of the inverse solution by introducing the return difference condition constraint for the parameter estimation and then solving the parameter estimation problem using constrained optimization.

The return difference condition guarantees that a *positive definite* inverse solution (with $\mathbf{Q}_0 \geq 0$ and $f > 0$) for the optimal control problem exists. The single input formulation was proposed in [73] and the general version in [8]. In terms of DBSFM, the single input return difference condition for the optimal controller can be stated as follows $\forall\omega \in \mathbb{R}$ must hold

$$C_r = G_r(i\omega)^H G_r(i\omega) - 1 \geq 0, \text{ where } G_r(i\omega) = 1 + \tilde{\mathbf{g}}_p \left(i\omega\mathbf{I} - \mathbf{A}_0\right)^{-1} \mathbf{B}_0, \tag{6.31}$$

and $\tilde{\mathbf{g}}_p = \begin{bmatrix}\mathbf{g}_p & 1/\tau_n\end{bmatrix}$. For a gradient-based constrained optimization algorithm, one needs the derivative of the return difference condition with respect to the unknown parameters. The gradient of (6.31) is given by

$$\frac{\partial C_r}{\partial \theta_l} = 2\mathrm{Re}\left(G_r(i\omega)^H \frac{\partial G_r(i\omega)}{\partial \theta_l}\right). \tag{6.32}$$

Because the optimal estimation and optimal control problems are dual (but transpose), the return difference condition for the estimator can easily be written using symmetry. The return difference for the estimation equations guaranteeing the existence of the *positive definite* inverse solution (with $\mathbf{W}_1 \geq 0$ and $\mathbf{V}_y > 0$) can be written as $\forall\omega \in \mathbb{R}$ must hold

$$C_f = G_f(i\omega)^H G_f(i\omega) - 1 \geq 0, \text{ where } G_f(i\omega) = 1 + \mathbf{C}_1 \left(i\omega\mathbf{I} - \mathbf{A}_1\right)^{-1} \mathbf{F}. \tag{6.33}$$

The gradient of the return difference for estimator is

$$\frac{\partial C_f}{\partial \theta_l} = 2\mathrm{Re}\left(G_f(i\omega)^H \frac{\partial G_f(i\omega)}{\partial \theta_l}\right). \tag{6.34}$$

In practice, the constraints (6.31) and (6.33) can be satisfied so that in each iteration, the minimum value as a function of the frequency is searched by solving the optimization problems

$$\omega_{\mathrm{r,min}} = \arg\min_w C_r(i\omega) \tag{6.35}$$

$$\omega_{\mathrm{f,min}} = \arg\min_w C_f(i\omega). \tag{6.36}$$

The problems (6.35) and (6.36) can be solved by using gradient-based optimization methods. However, because the return difference conditions might have several minima, the optimization should be carried out using several different initial values for ω. The gradients of the constraints

with respect to the frequency can be evaluated similarly as in (6.32) and (6.34). Now using (6.35) and (6.36) the criteria (6.35) and (6.36) can be simplified to

$$C_r(\omega_{\mathrm{r,min}}) \geq 0 \tag{6.37}$$
$$C_f(\omega_{\mathrm{f,min}}) \geq 0. \tag{6.38}$$

6.4.5 Inverse optimal control and estimation solutions for DBSFM

In order to adapt the human model to the changes in the operating point (as discussed in Section 6.1, the inverse optimal control and estimation problems should be solved. That is, "given the feedback gain matrix $\mathbf{g}_p$, the estimator gain matrix $\mathbf{F}$, the human operator time delay τ_p and the neuromotor time constaint τ_n, find the associated weight parameters f and $\mathbf{Q}_0$ as well as the covariances $\mathbf{V}_y$ and $\mathbf{W}_1$ so that $\mathbf{g}_p$, $\mathbf{F}$, τ_p and τ_n are the solutions of the associated MOCM-like LQG problem". The inverse optimal control problem was first considered by Kalman in [73], where the solution and conditions for the scalar case was presented. The inverse problem of optimal control can be solved as a singular value problem based on the return difference condition [8], but in this thesis LMIs are used, because the problem statement in the LMI domain is easy, the numerical solution can be found very efficiently, and there are readily available easy-to-use tools for programming LMI problems.

6.4.5.1 LMI solution for control weighting matrices

The general LMI approach for the inverse optimal control problem is presented in Appendix C.1. In the following, the solution is expressed in terms of DBSFM. Assume that the operator minimizes a general quadratic performance index modified from (6.16)

$$J_p = \lim_{t\to\infty} \frac{1}{t} \int_{t=0}^{t} \left(\boldsymbol{\chi}^T \mathbf{Q}_0 \boldsymbol{\chi} + \dot{u}_p^T f \dot{u}_p + 2\boldsymbol{\chi}^T \mathbf{N}_0 \dot{u}_p \right) dt \tag{6.39}$$

where $\mathbf{N}_0 \in \mathbb{R}^{(N_a+1)\times 1}$, $\mathbf{Q}_0 \in \mathbb{R}^{(N_a+1)\times(N_a+1)}$, and $f \in \mathbb{R}$ are unknown variables. The cross term weighting $\mathbf{N}_0 = 0$ if the return difference condition (6.31) is fulfilled. The solution of (6.39) is obtained using the 0-plant (B.8). The optimal control law is

$$\dot{u}_p = -\hat{\tilde{\mathbf{g}}}_p \boldsymbol{\chi} = -f^{-1} \left(\mathbf{B}_0^T \mathbf{K} + \mathbf{N}_0^T \right) \hat{\boldsymbol{\chi}}, \tag{6.40}$$

where $\mathbf{K}$ is the positive definite solution of the Riccati equation

$$0 = \mathbf{A}_0^T \mathbf{K} + \mathbf{K}\mathbf{A}_0 + \mathbf{Q}_0 - \left(\mathbf{B}_0^T \mathbf{K} + \mathbf{N}_0^T \right)^T f^{-1} \left(\mathbf{B}_0^T \mathbf{K} + \mathbf{N}_0^T \right), \tag{6.41}$$

when $f > 0$ and $\mathbf{Q}_0 - \mathbf{N}_0 f^{-1} \mathbf{N}_0^T \geq 0$. The solution of the inverse optimal control problem will result in parameters $\mathbf{K}$, $\mathbf{Q}_0$, f and $\mathbf{N}_0$, which will produce a gain $\hat{\tilde{\mathbf{g}}}_p$ that is equal to the gain $\tilde{\mathbf{g}}_p$ given by the DBSFM identification. Note that introducing a nonzero crossterm $\mathbf{N}_0$ turns (6.39) nonconvex, because $\boldsymbol{\chi}$ and $\dot{u}_p$ can have different signs. Thus, the inverse solution will not be unique and can converge to a local optimum.

Defining $\tilde{\mathbf{g}}_p = \begin{bmatrix} \mathbf{g}_p & \frac{1}{\tau_n} \end{bmatrix}$, the unknown weighting matrices $\mathbf{Q}_0$, f and $\mathbf{N}_0$ can be obtained by

solving the LMI problem

$$\begin{cases} \min \lambda \text{ s.t.} \\ \begin{cases} \mathbf{A}_0^T\mathbf{K} + \mathbf{K}\mathbf{A}_0 - \tilde{\mathbf{g}}_p^T f \tilde{\mathbf{g}}_p \leq 0 \\ \begin{bmatrix} \mathbf{Z} & \mathbf{S}^T \\ \mathbf{S} & \mathbf{I} \end{bmatrix} > 0, \text{where } \mathbf{S}^T = \mathbf{B}^T\mathbf{K} + \mathbf{N}_0^T - \tilde{\mathbf{g}}_p f \\ \mathbf{Z} < \lambda \mathbf{I}, \ \begin{bmatrix} \mathbf{Q}_0 & \mathbf{N}_0 \\ \mathbf{N}_0^T & f \end{bmatrix} \geq 0, \\ \text{where } \mathbf{Q}_0 = -\mathbf{A}_0^T\mathbf{K} - \mathbf{K}\mathbf{A}_0 + \tilde{\mathbf{g}}_p^T f \tilde{\mathbf{g}}_p \end{cases} \end{cases} \tag{6.42}$$

with $\mathbf{K} > 0$ and $f > 0$. Formulated as such, the problem (6.42) can readily be solved as a Generalized Eigenvalue Problem (GEVP) using the standard LMI tools.

6.4.5.2 LMI solution for estimator weighting matrices

For the Kalman filter derivation the following modification of the 1-plant (B.16) is considered ($\mathbf{E}_1 = \mathbf{I}$)

$$\begin{cases} \dot{\boldsymbol{\chi}} = \mathbf{A}_1\boldsymbol{\chi} + \mathbf{B}_1 u_c + \mathbf{w}_1, \text{ where } \mathbf{w}_1 \sim (0, \mathbf{W}_1) \\ \mathbf{y}_{\text{obs}} = \mathbf{C}_1\boldsymbol{\chi} + \mathbf{v}_y, \text{ where } \mathbf{v}_y \sim (0, \mathbf{V}_y) \end{cases}, \tag{6.43}$$

where the expectation $\mathrm{E}\left[\mathbf{w}_1\mathbf{v}_y^T\right] = \mathbf{N}_1 \in \mathbb{R}^{(N_a+1)\times N_y}$, and the initial values are as $\boldsymbol{\chi}(0) = \boldsymbol{\chi}_0$. Now $\mathbf{N}_1$, $\mathbf{W}_1 \in \mathbb{R}^{(N_a+1)\times(N_a+1)}$, and $\mathbf{V}_y \in \mathbb{R}^{(N_y)\times(N_y)}$ are unknown matrix variables. If the return difference condition (6.33) is fulfilled, the inverse optimal solution with $\mathbf{N}_1 = 0$ exists. The gain $\mathbf{F}$ of the steady state Kalman filter for the process (6.43) is given by [141]

$$\hat{\mathbf{F}} = \left(\boldsymbol{\Sigma}_1\mathbf{C}_1^T + \mathbf{N}_1\right)\mathbf{V}_y^{-1}, \tag{6.44}$$

where $\boldsymbol{\Sigma}_1$ is the positive definite solution of the algebraic Riccati equation

$$0 = \mathbf{A}_1\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_1\mathbf{A}_1^T + \mathbf{W}_1 - \left(\boldsymbol{\Sigma}_1\mathbf{C}_1^T + \mathbf{N}_1\right)\mathbf{V}_y^{-1}\left(\boldsymbol{\Sigma}_1\mathbf{C}_1^T + \mathbf{N}_1\right)^T \tag{6.45}$$

The gain matrix of the state estimator given by the DBSFM identification is $\mathbf{F}$. The objective of the inverse optimal estimation problem is to find such parameters $\boldsymbol{\Sigma}_1$, $\mathbf{N}_1$, $\mathbf{V}_y$ and $\mathbf{W}_1$ which will produce the gain $\hat{\mathbf{F}}$ that is equal to $\mathbf{F}$ given by the DBSFM identification. The solution can be obtained in a similar manner as the inverse optimal control solution.

The noise intensity matrices $\mathbf{V}_y$ and $\mathbf{W}_1$ which result in the corresponding gain can be obtained by solving the following LMI problem.

$$\begin{cases} \min \lambda \text{ s.t.} \\ \begin{cases} \mathbf{A}_1\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_1\mathbf{A}_1^T - \mathbf{F}\mathbf{V}_y\mathbf{F}^T \leq 0 \\ \begin{bmatrix} \mathbf{Z} & \mathbf{S}^T \\ \mathbf{S} & \mathbf{I} \end{bmatrix} > 0, \text{where } \mathbf{S} = \boldsymbol{\Sigma}_1\mathbf{C}_1^T + \mathbf{N}_1 - \mathbf{F}\mathbf{V}_y \\ \mathbf{Z} < \lambda \mathbf{I}, \ \begin{bmatrix} \mathbf{W}_1 & \mathbf{N}_1 \\ \mathbf{N}_1^T & \mathbf{V}_y \end{bmatrix} \geq 0, \\ \text{where } \mathbf{W}_1 = -\mathbf{A}_1\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_1\mathbf{A}_1^T + \mathbf{F}\mathbf{V}_y\mathbf{F}^T \end{cases} \end{cases} \tag{6.46}$$

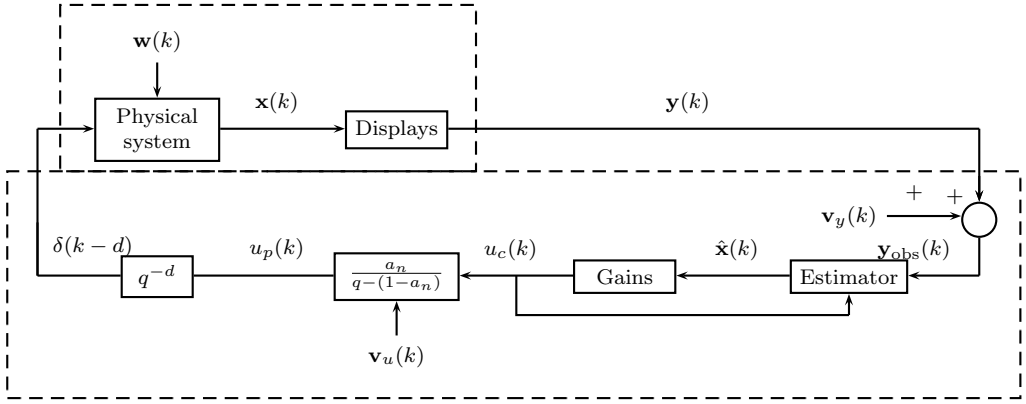


Figure 6.4: The structure of DDBSFM model. The blocks inside the dashed box including neuromotor lag, gains, estimator, and delay describe the human operator. The combination of the physical system and the displays comprises the plant the operator controls.

where $\mathbf{\Sigma}_1 > 0$ and $\mathbf{V}_y > 0$. Similarly as the inverse optimal control problem (6.42) also (6.46) can be solved as GEVP.

6.5 Discrete data-based state feedback model of human operator

6.5.1 General structure

The structure of the Discrete Data-Based State Feedback Model (DDBSFM) is shown in Figure 6.4, which describes the multi-input single output (MISO) DDBSFM. The structure is similar to the MOCM and the DBSFM, but the state feedback and the estimator gains, as well as the neuromotor dynamics and the dead time lag are derived in discrete time domain. The dead time delay τ_p is assumed to be a multiplier of the sampling time h.

Consider the 0-plant (B.8). By applying the ZOH discretization with the step size h, the discrete-time equivalent 0-plant becomes

$$\begin{cases} \boldsymbol{\chi}(k+1) = \mathbf{\Phi}_0 \boldsymbol{\chi}(k) + \bar{\mathbf{\Gamma}}_0 \frac{\Delta u_p(k+1)}{h} + \mathbf{\Gamma}_w \mathbf{w}(k) \\ \mathbf{y}_{\text{obs}}(k) = \mathbf{C}_0 \boldsymbol{\chi}(k) + \mathbf{v}_y(k) \end{cases}, \tag{6.47}$$

where the matrices are $\mathbf{\Phi}_0 = e^{\mathbf{A}_0 h}$, $\bar{\mathbf{\Gamma}}_0 = \int_0^h e^{\mathbf{A}_0 \lambda} \mathbf{B}_0 d\lambda$, $\mathbf{\Gamma}_w = \int_0^h e^{\mathbf{A}_0 \lambda} \mathbf{E}_0 d\lambda$, and $\frac{\Delta u_p(k+1)}{h} = \frac{u_p(k+1) - u_p(k)}{h}$ is the finite difference approximation of the derivative $\dot{u}_p$ at $k+1$. The state vector is defined as $\boldsymbol{\chi}^T(k) = \begin{bmatrix} \mathbf{x}^T(k) & \mathbf{x}_d^T(k) & u_p(k) \end{bmatrix}$ and the initial values of (6.47) are $\boldsymbol{\chi}_0$ as in (B.8). Note that the delay τ_p could have as well be expressed by augmenting the delayed states into the state vector, as it is usually done in discrete time systems. However, using Padé approximation retains the size of the 0-plant as $N_s + 2 + 1$, which is more convenient.

Now let us take the expression of $\dot{u}_p$ in (B.8) under more careful study. Defining $\dot{u}_p(k+1) \approx$

$\frac{\Delta u_p(k+1)}{h} = \frac{u_p(k+1)-u_p(k)}{h}$, the discretization of

$$\dot{u}_p = 0u_p + 1\dot{u}_p \tag{6.48}$$

leads to the incremental expression

$$\begin{aligned} u_p(k+1) &= e^{0h}u_p(k) + \int_0^h e^{0\lambda}1d\lambda\frac{\Delta u_p(k+1)}{h} \\ &= u_p(k) + h\frac{\Delta u_p(k+1)}{h} \\ &= u_p(k) + \Delta u_p(k+1). \end{aligned} \tag{6.49}$$

Now the discrete-time 0-plant can be written as

$$\begin{cases} \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}_d(k+1) \\ u_p(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_0^{1,1} & \mathbf{\Phi}_0^{1,2} & \mathbf{\Phi}_0^{1,3} \\ 0 & \mathbf{\Phi}_0^{2,2} & \mathbf{\Phi}_0^{2,3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_d(k) \\ u_p(k) \end{bmatrix} + \begin{bmatrix} h\mathbf{\Gamma}_0^{1,1} \\ h\mathbf{\Gamma}_0^{2,1} \\ 1 \end{bmatrix} \Delta u_p(k+1) + \begin{bmatrix} \mathbf{\Gamma}_{0w}^{1,1} \\ \mathbf{\Gamma}_{0w}^{2,1} \\ \mathbf{\Gamma}_{0w}^{3,1} \end{bmatrix} \mathbf{w}(k) \\ \mathbf{y}_{\text{obs}} = \begin{bmatrix} \mathbf{C}\ \mathbf{D}\mathbf{C}_d\ \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_d(k) \\ u_p(k) \end{bmatrix} + \mathbf{v}_y(k) \end{cases}, \tag{6.50}$$

where $\mathbf{\Phi}_0^{i,j}$, $\mathbf{\Gamma}_0^i$ and $\mathbf{\Gamma}_{0w}^i$ are the correponding blocks of the matrices $\mathbf{\Phi}_0$, $\mathbf{\Gamma}_0$ and $\mathbf{\Gamma}_{0w}$ in (6.47). Since $\boldsymbol{\chi}^T(k) = \begin{bmatrix} \mathbf{x}^T(k) & \mathbf{x}_d^T(k) & u_p(k) \end{bmatrix}$ the discrete time 0-plant can be written compactly as

$$\begin{cases} \boldsymbol{\chi}(k+1) = \mathbf{\Phi}_0\boldsymbol{\chi}(k) + \mathbf{\Gamma}_0\Delta u_p(k+1) + \mathbf{\Gamma}_{0w}\mathbf{w}(k) \\ \mathbf{y}_{\text{obs}}(k) = \mathbf{C}_0\boldsymbol{\chi}(k) + \mathbf{v}_y(k) \end{cases}, \tag{6.51}$$

where the initial values are $\boldsymbol{\chi}(0) = \boldsymbol{\chi}_0$, and matrix notations are obvious.

The 0-plant is controlled by the increment $\Delta u_p(k+1)$. Therefore, the state feedback control law becomes

$$\Delta u_p(k+1) = -\tilde{\mathbf{g}}_p\hat{\boldsymbol{\chi}}(k) = -\begin{bmatrix} g_1 \dots g_{N_a}\ g_{N_a+1} \end{bmatrix}\hat{\boldsymbol{\chi}}(k). \tag{6.52}$$

Defining $\mathbf{g}_p = \begin{bmatrix} g_1 \dots g_{N_a} \end{bmatrix}$ and using $\hat{\boldsymbol{\chi}}(k)$ the control law (6.52) can be written as

$$\Delta u_p(k+1) = -\mathbf{g}_p \begin{bmatrix} \hat{\mathbf{x}}(k) \\ \hat{\mathbf{x}}_d(k) \end{bmatrix} - g_{N_a+1}\hat{u}_p(k) \tag{6.53}$$

Moreover, by requiring $g_{N_a+1} = a_n$ and $\mathbf{l}_p = \frac{\mathbf{g}_p}{a_n}$ the control law becomes

$$\Delta u_p(k+1) = a_n u_c(k) - a_n u_p(k), \tag{6.54}$$

where

$$u_c(k) = -\mathbf{l}_p \begin{bmatrix} \hat{\mathbf{x}}(k) \\ \hat{\mathbf{x}}_d(k) \end{bmatrix} \tag{6.55}$$

Assuming that the operator's control signal is corrupted by zero-mean white noise $v_u(k)$ with

intensity V_u, equation (6.54) becomes

$$\Delta u_p(k+1) = a_n u_c(k) - a_n u_p(k) + a_n v_u(k). \tag{6.56}$$

Now inserting (6.56) into (6.51) results in

$$\begin{cases} \begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}_d(k+1) \\ u_p(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_0^{1,1} & \mathbf{\Phi}_0^{1,2} & \mathbf{\Phi}_0^{1,3} \\ 0 & \mathbf{\Phi}_0^{2,2} & \mathbf{\Phi}_0^{2,3} \\ 0 & 0 & 1-a_n \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_d(k) \\ u_p(k) \end{bmatrix} + \begin{bmatrix} h\mathbf{\Gamma}_0^{1,1} \\ h\mathbf{\Gamma}_0^{2,1} \\ a_n \end{bmatrix} u_c(k) + \begin{bmatrix} \mathbf{\Gamma}_{0w}^{1,1} & 0 \\ \mathbf{\Gamma}_{0w}^{2,1} & 0 \\ \mathbf{\Gamma}_{0w}^{3,1} & a_n \end{bmatrix} \begin{bmatrix} \mathbf{w}(k) \\ v_u(k) \end{bmatrix} \\ \mathbf{y}_{\text{obs}} = \begin{bmatrix} \mathbf{C} & \mathbf{D}\mathbf{C}_d & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_d(k) \\ u_p(k) \end{bmatrix} + \mathbf{v}_y(k) \end{cases}, \tag{6.57}$$

which can be written compactly as

$$\begin{cases} \boldsymbol{\chi}(k+1) = \mathbf{\Phi}_1 \boldsymbol{\chi}(k) + \mathbf{\Gamma}_1 u_c(k) + \mathbf{\Gamma}_{1w} \mathbf{w}_1(k) \\ \mathbf{y}_{\text{obs}}(k) = \mathbf{C}_1 \boldsymbol{\chi}(k) + \mathbf{v}_y(k) \end{cases} \tag{6.58}$$

with obvious matrix notations and the initial values defined as $\boldsymbol{\chi}(0) = \boldsymbol{\chi}_0$. In terms of MOCM, this is the 1-plant, which is used to synthesize the state estimator of the human operator.

By using the 1-plant, the state estimator of the human operator assumes the form

$$\hat{\boldsymbol{\chi}}(k+1) = \mathbf{\Phi}_1 \hat{\boldsymbol{\chi}}(k) + \mathbf{\Gamma}_1 u_c(k) + \mathbf{F}\left(\mathbf{y}_{\text{obs}}(k) - \hat{\mathbf{y}}(k)\right), \tag{6.59}$$

where $\mathbf{y}_{\text{obs}}(k) = \mathbf{y}(k) + \mathbf{v}_y(k)$ and $\hat{\mathbf{y}}(k) = \mathbf{C}_1 \hat{\boldsymbol{\chi}}(k)$. The measurable output of the controlled plant (B.1) (after applying ZOH discretization) is denoted by $\mathbf{y}(k)$. Thus, equation (6.59) becomes

$$\hat{\boldsymbol{\chi}}(k+1) = \left(\mathbf{\Phi}_1 - \mathbf{F}\mathbf{C}_1\right) \hat{\boldsymbol{\chi}}(k) + \mathbf{F}\mathbf{y}(k) + \mathbf{\Gamma}_1 u_c(k) + \mathbf{F}\mathbf{v}_y(k) \tag{6.60}$$

where $\mathbf{F}$ is the unknown estimator gain and $\hat{\boldsymbol{\chi}}^T(k) = \begin{bmatrix} \hat{\mathbf{x}}^T(k) & \hat{\mathbf{x}}_d^T(k) & \hat{u}_p(k) \end{bmatrix}$ is the state estimate.

Defining $\mathbf{x}_p^T = \begin{bmatrix} \hat{\mathbf{x}}^T & \hat{\mathbf{x}}_d^T & \hat{u}_p & u_p \end{bmatrix}$, $\mathbf{l}_1 = \begin{bmatrix} \frac{\mathbf{g}_p}{a_n} & 0 \end{bmatrix}$ and using (6.55), (6.56), and (6.58), the human operator's dynamics can be written as

$$\begin{cases} \mathbf{x}_p(k+1) = \begin{bmatrix} \mathbf{\Phi}_1 - \mathbf{F}\mathbf{C}_1 - \mathbf{\Gamma}_0 \begin{bmatrix} \mathbf{g}_p & 0 \end{bmatrix} & 0 \\ -\begin{bmatrix} \mathbf{g}_p & 0 \end{bmatrix} & 1-a_n \end{bmatrix} \mathbf{x}_p(k) + \begin{bmatrix} \mathbf{F} \\ 0 \end{bmatrix} \mathbf{y}(k) + \begin{bmatrix} \mathbf{F} & 0 \\ 0 & a_n \end{bmatrix} \begin{bmatrix} \mathbf{v}_y(k) \\ v_u(k) \end{bmatrix} \\ \delta(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}_p(k-d) \end{cases}.$$

or more compactly

$$\begin{cases} \mathbf{x}_p(k+1) = \mathbf{\Phi}_p \mathbf{x}_p(k) + \mathbf{\Gamma}_p \mathbf{y}(k) + \mathbf{\Gamma}_v \mathbf{v}_p(k) \\ \delta(k) = \mathbf{C}_p \mathbf{x}_p(k-d) \end{cases} \tag{6.61}$$

with obvious matrix notations and the initial values as $\mathbf{x}_p(0) = \mathbf{x}_{p,0}$ where the vector $\mathbf{x}_{p,0}^T = \begin{bmatrix} \hat{\mathbf{x}}^T(0) & \hat{\mathbf{x}}_d^T(0) & \hat{u}_p(0) & u_p(0) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_0^T & \hat{\mathbf{x}}_{d,0}^T & \hat{u}_{p,0} & u_{p,0} \end{bmatrix}$. The unknown parameters of the model are the state estimator gain matrix $\mathbf{F}$, the state feedback gain vector $\mathbf{g}_p$ and the neuromotor constant a_n. In addition, the delay d is assumed to be known beforehand.

6.5.2 DDBSFM parameter estimation algorithm

6.5.2.1 Cost function and gradient

Consider again the cost function

$$V(\theta) = \frac{1}{2}\sum_{k=1}^{N}\left(\delta_m(k) - \delta(k)\right)^2, \tag{6.62}$$

where $\delta_m(k)$ is the true human control signal, $\delta(k)$ the signal given by the DDBSFM, and $\theta \in \mathbb{R}^{N_\theta}$ the vector of the unknown parameters. The minimization of the cost function can be performed using the gradient-based optimization algorithms. By using the chain rule, the gradient of (6.62) with respect to the l^{th} component of the unknown parameter vector becomes

$$\frac{\partial V(\theta)}{\partial \theta_l} = -\sum_{k=1}^{N}\left(\delta_m(k) - \delta(k)\right)\frac{\partial \delta(k)}{\partial \theta_l}. \tag{6.63}$$

Assuming that $\mathbf{C}_p$ does not depend on θ_l, the derivative of the output $\delta(k)$ can be solved by applying again the chain rule for the output of (6.61) as

$$\frac{\partial \delta(k)}{\partial \theta_l} = \mathbf{C}_p\frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_l}. \tag{6.64}$$

Because $\mathbf{x}_p$ depends on θ_l, the derivative $\frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_l}$ is obtained by applying the chain rule for the state equation of (6.61) as follows.

$$\frac{\partial \mathbf{x}_p(k+1)}{\partial \theta_l} = \frac{\partial \mathbf{\Phi}_p}{\partial \theta_l}\mathbf{x}_p(k) + \mathbf{\Phi}_p\frac{\partial \mathbf{x}_p(k)}{\partial \theta_l} + \frac{\partial \mathbf{\Gamma}_p}{\partial \theta_l}\mathbf{y}(k) \tag{6.65}$$

As a result, the gradient becomes a dynamical system where the state vector is $\frac{\partial \mathbf{x}_p}{\partial \theta_l}$ and the inputs are $\mathbf{x}_p(k)$ and $\mathbf{y}(k)$. In this thesis (6.65) is called the Gradient Propagation Difference Equation (GPDE), which is needed for each component of θ. The initial values of the GPDE are zero, because the initial values of the human operator states $\mathbf{x}_p(0)$ are constant.

6.5.3 Initial values for the gradient-based algorithm

In DDBSFM the dead time delay lag d should be known beforehand. In practice, one can try to identify the model parameters by using different delay values and then choose the best. The initial value for a_n should be chosen from $0 < a_n < 1$.

If one assumes that the human operator controls the process as an optimal controller minimizing the performance index

$$J_p = \lim_{N\to\infty}\frac{1}{N}\sum_{k=0}^{N}\left(\boldsymbol{\chi}^T(k)\mathbf{Q}_0\boldsymbol{\chi}(k) + \Delta u_p^T(k+1) f \Delta u_p(k+1)\right), \tag{6.66}$$

the discrete Linear Quadratic Gaussian (LQG) solution for the control law becomes

$$\begin{aligned}\Delta u_p(k+1) &= -\tilde{\mathbf{g}}_p\hat{\boldsymbol{\chi}}(k) = -\left[g_1 \dots g_{N_a}\, g_{N_a+1}\right]\hat{\boldsymbol{\chi}}(k) \\ &= -\left(\boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Gamma}_0 + f\right)^{-1}\boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Phi}_0\hat{\boldsymbol{\chi}}(k),\end{aligned} \tag{6.67}$$

where $\mathbf{K}$ is the positive-definite solution of the Riccati equation

$$\mathbf{K} = \boldsymbol{\Phi}_0^T\mathbf{K}\boldsymbol{\Phi}_0 - \boldsymbol{\Phi}_0^T\mathbf{K}\boldsymbol{\Gamma}_0\left(\boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Gamma}_0 + f\right)^{-1}\boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Phi}_0 + \mathbf{Q}_0, \tag{6.68}$$

with $\mathbf{Q}_0 \geq 0$ and $f \geq 0$. Moreover, it is required that $a_n = g_{N_a+1}$. Therefore, one first decides the initial guess for a_n. Then the solution of the initial value of the feedback gain matrix requires the solution of the optimization problem

$$\min_f \left(g_{N_a+1} - a_n\right)^2, \tag{6.69}$$

when g_{N_a+1} is as in (6.67) and f as in (6.68). The weighting matrix $\mathbf{Q}_0$ can be chosen based on the prior knowledge of the task the operator is trying to execute. Similarly as with (B.40) the problem (6.69) can be solved by defining the LMIs

$$\mathbf{L}_1 = \begin{bmatrix}\boldsymbol{\Phi}_0^T\mathbf{K}\boldsymbol{\Phi}_0 - \mathbf{K} + \mathbf{Q}_0 & \boldsymbol{\Phi}_0^T\mathbf{K}\boldsymbol{\Gamma}_0 \\ \boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Phi}_0 & \boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Gamma}_0 + f\end{bmatrix} \geq 0 \tag{6.70}$$

$$\mathbf{L}_2 = \begin{bmatrix}\lambda & \left(\boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Phi}_0\mathbf{D}_1 - \left(\boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Phi}_0 + f\right)a_n\right)^T \\ \boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Phi}_0\mathbf{D}_1 - \left(\boldsymbol{\Gamma}_0^T\mathbf{K}\boldsymbol{\Phi}_0 + f\right)a_n & 1\end{bmatrix} \geq 0, \tag{6.71}$$

where $\mathbf{D}_1 = \begin{bmatrix}0 & 0 & \dots & 1\end{bmatrix}^T$ ($\mathbf{D}_1 \in \mathbb{R}^{N_a+1}$), and solving the LMI problem (LMIP)

$$\begin{aligned}&\min_{f,\mathbf{K},\lambda}\ w_\lambda\lambda - w_K\mathrm{tr}\left(\mathbf{K}\right) \\ &\text{s.t. } \mathbf{L}_1 \geq 0, \mathbf{L}_2 \geq 0, \mathbf{K} \geq 0, f > 0.,\end{aligned} \tag{6.72}$$

where w_λ and w_K are positive weights. Again one needs to define weights so that $w_\lambda \gg w_K$.

If one assumes that (6.60) is a Kalman filter, the estimator gain matrix can be obtained by solving

$$\mathbf{F} = \boldsymbol{\Phi}_1\boldsymbol{\Sigma}_1\mathbf{C}_1\left(\mathbf{C}_1\boldsymbol{\Sigma}_1\mathbf{C}_1^T + \mathbf{V}_y\right)^{-1}, \tag{6.73}$$

where $\boldsymbol{\Sigma}_1$ is the positive definite solution of the Riccati equation

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Phi}_1\boldsymbol{\Sigma}_1\boldsymbol{\Phi}_1^T - \boldsymbol{\Phi}_1\boldsymbol{\Sigma}_1\mathbf{C}_1\left(\mathbf{C}_1\boldsymbol{\Sigma}_1\mathbf{C}_1^T + \mathbf{V}_y\right)^{-1}\mathbf{C}_1\boldsymbol{\Sigma}_1\boldsymbol{\Phi}_1^T + \mathbf{W}_1, \tag{6.74}$$

with $\mathbf{W}_1 \geq 0$ and $\mathbf{V}_y > 0$. The initial guess for the state estimator matrix $\mathbf{F}$ can be obtained by solving (6.73). Choosing $\mathbf{W}_1$ and $\mathbf{V}_y$ as identity matrices of appropriate dimensions will give a suitable initial value for $\mathbf{F}$.

6.5.4 Suboptimal inverse control and estimation solutions for DDBSFM

6.5.4.1 Suboptimal LMI solution for discrete-time control weighting matrices

Similarly as with the continuous DBSFM, the DDBSFM can be converted to the optimal control framework by solving the associated inverse problem. The LMI approach for the discrete time inverse optimal control problem is described in Appendix C.2. In the following, the modified version for DDBSFM is presented. Assume that the operator minimizes a general quadratic performance index

$$J_p = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} \left(\boldsymbol{\chi}^T(k) \mathbf{Q}_0 \boldsymbol{\chi}(k) + \Delta u_p^T(k+1) f \Delta u_p(k+1) + 2\boldsymbol{\chi}^T(k) \mathbf{N}_0 \Delta u_p(k+1) \right) \tag{6.75}$$

which includes the state-control crossterm $2\hat{\boldsymbol{\chi}}^T(k)\mathbf{N}_0\Delta u_p(k+1)$ (the dimension of is $\mathbf{N}_0 \in \mathbb{R}^{(N_a+1)\times 1}$) because the DDBSFM identification can result in parameters which are not solutions of the standard optimal control performance index. As pointed out in [8, 165], the return difference condition for the discrete time optimal control systems cannot be expressed in a simple frequency domain form as in the continuous case. Thus, only the suboptimal inverse optimal solution for DDBSFM where the crossterm is included is given in this thesis.

The solution of (6.75) is obtained using the 0-plant (6.51). The optimal control law is

$$\Delta u_p(k+1) = -\hat{\tilde{\mathbf{g}}}_p \hat{\boldsymbol{\chi}}(k) = -\left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Gamma}_0 + f \right)^{-1} \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Phi}_0 + \mathbf{N}_0^T \right) \hat{\boldsymbol{\chi}}(k), \tag{6.76}$$

where $\mathbf{K}$ is the positive definite solution of the Riccati equation

$$\mathbf{K} = \boldsymbol{\Phi}_0^T \mathbf{K} \boldsymbol{\Phi}_0 - \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Phi}_0 + \mathbf{N}_0^T \right)^T \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Gamma}_0 + f \right)^{-1} \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Phi}_0 + \mathbf{N}_0^T \right) + \mathbf{Q}_0, \tag{6.77}$$

when $f > 0$ and $\mathbf{Q}_0 - \mathbf{N}_0 f^{-1} \mathbf{N}_0^T \geq 0$. The solution of the inverse optimal control problem will result in parameters $\mathbf{K}$, $\mathbf{Q}_0$, f and $\mathbf{N}_0$, which will produce a gain $\hat{\tilde{\mathbf{g}}}_p$ that is equal to the gain $\tilde{\mathbf{g}}_p$ given by the DBSFM identification.

Defining $\tilde{\mathbf{g}}_p = \begin{bmatrix} \mathbf{g}_p & a_n \end{bmatrix}$, the unknown weighting matrices $\mathbf{Q}_0$, f and $\mathbf{N}_0$ can be obtained by solving the LMI problem

$$\begin{aligned} &\min \lambda \text{ s.t.} \\ &\begin{cases} \boldsymbol{\Phi}_0^T \mathbf{K} \boldsymbol{\Phi}_0 - \tilde{\mathbf{g}}_p^T \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Gamma}_0 + f \right) \tilde{\mathbf{g}}_p - \mathbf{K} \leq 0 \\ \begin{bmatrix} \mathbf{Z} & \mathbf{S}^T \\ \mathbf{S} & \mathbf{I} \end{bmatrix} > 0, \text{where } \mathbf{S}^T = \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Gamma}_0 + f \right) \tilde{\mathbf{g}}_p - \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Phi}_0 + \mathbf{N}_0^T \right) \\ \mathbf{Z} < \lambda \mathbf{I}, \ \begin{bmatrix} \mathbf{Q}_0 & \mathbf{N}_0 \\ \mathbf{N}_0^T & f \end{bmatrix} \geq 0, \\ \text{where } \mathbf{Q}_0 = -\boldsymbol{\Phi}_0^T \mathbf{K} \boldsymbol{\Phi}_0 + \tilde{\mathbf{g}}_p^T \left(\boldsymbol{\Gamma}_0^T \mathbf{K} \boldsymbol{\Gamma}_0 + f \right) \tilde{\mathbf{g}}_p + \mathbf{K} \end{cases} \end{aligned} \tag{6.78}$$

with $\mathbf{K} > 0$ and $f > 0$. Formulated as such, the problem (6.78) can readily be solved as a GEVP using the standard LMI tools.

6.5.4.2 Suboptimal LMI solution for discrete-time estimator weighting matrices

For obtaining the Kalman filter parameters using the inverse optimal solution, the following modification of the discrete-time 1-plant (6.58) is considered ($\mathbf{\Gamma}_{1w} = \mathbf{I}$)

$$\begin{cases} \boldsymbol{\chi}(k+1) = \mathbf{\Phi}_1\boldsymbol{\chi}(k) + \mathbf{\Gamma}_1 u_c(k) + \mathbf{w}_1(k), \text{ where } \mathbf{w}_1(k) \sim (0, \mathbf{W}_1) \\ \mathbf{y}_{\text{obs}}(k) = \mathbf{C}_1\boldsymbol{\chi}(k) + \mathbf{v}_y(k), \text{ where } \mathbf{v}_y(k) \sim (0, \mathbf{V}_y) \end{cases}, \tag{6.79}$$

where the expectation $\mathrm{E}\left[\mathbf{w}_1(k)\mathbf{v}_y^T(k)\right] = \mathbf{N}_1$. Because the existence of the inverse solution with $\mathbf{N}_1 = 0$ and $\mathbf{W}_1 \geq 0$ cannot be guaranteed, only the suboptimal solution can be found.

The gain $\mathbf{F}$ of the steady state Kalman filter for the process (6.79) is given by [141]

$$\mathbf{F} = \left(\mathbf{\Phi}_1\mathbf{\Sigma}_1\mathbf{C}_1^T + \mathbf{N}_1\right)\left(\mathbf{C}_1\mathbf{\Sigma}_1\mathbf{C}_1^T + \mathbf{V}_y\right)^{-1}, \tag{6.80}$$

where $\mathbf{\Sigma}_1$ is the positive definite solution of the algebraic Riccati equation

$$\mathbf{\Sigma}_1 = \mathbf{\Phi}_1\mathbf{\Sigma}_1\mathbf{\Phi}_1^T - \mathbf{F}\left(\mathbf{C}_1\mathbf{\Sigma}_1\mathbf{C}_1^T + \mathbf{V}_y\right)\mathbf{F}^T + \mathbf{W}_1, \tag{6.81}$$

The gain matrix of the state estimator given by the DDBSFM identification is $\mathbf{F}$. The objective of the inverse optimal estimation problem is to find such parameters $\mathbf{\Sigma}_1$, $\mathbf{N}_1$, $\mathbf{V}_y$ and $\mathbf{W}_1$ which will produce gain $\hat{\mathbf{F}}$ when using (6.80) that is equal to $\mathbf{F}$ given by the DDBSFM identification. The solution can be obtained in a similar manner as the inverse optimal control solution.

The noise intensity matrices $\mathbf{V}_y$ and $\mathbf{W}_1$ which result in the corresponding gain can be obtained by solving the following LMI problem.

$$\begin{aligned} &\min \lambda \text{ s.t.} \\ &\begin{cases} \mathbf{\Phi}_1\mathbf{\Sigma}_1\mathbf{\Phi}_1^T - \mathbf{F}\left(\mathbf{C}_1\mathbf{\Sigma}_1\mathbf{C}_1^T + \mathbf{V}_y\right)\mathbf{F}^T - \mathbf{\Sigma}_1 \leq 0 \\ \begin{bmatrix} \mathbf{Z} & \mathbf{S}^T \\ \mathbf{S} & \mathbf{I} \end{bmatrix} > 0, \text{where} \\ \mathbf{S} = \mathbf{F}\left(\mathbf{C}_1\mathbf{\Sigma}_1\mathbf{C}_1^T + \mathbf{V}_y\right) - \left(\mathbf{\Phi}_1\mathbf{\Sigma}_1\mathbf{C}_1^T + \mathbf{N}_1\right) \\ \mathbf{Z} < \lambda\mathbf{I}, \ \begin{bmatrix} \mathbf{W}_1 & \mathbf{N}_1 \\ \mathbf{N}_1^T & \mathbf{V}_y \end{bmatrix} \geq 0, \text{where} \\ \mathbf{W}_1 = -\left(\mathbf{\Phi}_1\mathbf{\Sigma}_1\mathbf{\Phi}_1^T - \mathbf{F}\left(\mathbf{C}_1\mathbf{\Sigma}_1\mathbf{C}_1^T + \mathbf{V}_y\right)\mathbf{F}^T - \mathbf{\Sigma}_1\right) \end{cases} \end{aligned} \tag{6.82}$$

where $\mathbf{\Sigma}_1 > 0$ and $\mathbf{V}_y > 0$. Similarly as the inverse optimal control problem (6.78) also (6.82) can be solved as GEVP.

6.6 Hammerstein-Wiener formulation of DDBSFM

6.6.1 General structure

The general structure of the Hammerstein-Wiener DDBSFM (HWDDBSFM) is similar to the HWMOCM shown in Figure 6.3, where the MOCM core model is replaced with the DDBSFM

core model. The state space representation of the HWDDBSFM is

$$\begin{cases} \mathbf{x}_p(k+1) = \mathbf{\Phi}_p \mathbf{x}_p(k) + \mathbf{\Gamma}_p \mathcal{G}\left(\mathbf{y}(k)\right) + \mathbf{\Gamma}_v \mathbf{v}_p(k) \\ \delta(k) = \mathcal{F}\left(\mathbf{C}_p \mathbf{x}_p(k-d)\right) \end{cases}, \tag{6.83}$$

where the initial values are $\mathbf{x}_p(0) = \mathbf{x}_{p,0}$ as in (6.61).

6.6.2 Identification of the unknown parameters

In the HWDDBSFM case the unknown parameters can be identified by using the gradient-based optimization algorithms. Assuming that $\mathbf{C}_p$ does not depend on θ_l, the derivative of the output $\delta(k)$ is obtained using the chain rule as follows

$$\frac{\partial \delta(k)}{\partial \theta_l} = \frac{\partial \mathcal{F}\left(\mathbf{C}_p \mathbf{x}_p(k-d)\right)}{\partial \mathbf{x}_p(k-d)} \frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_l}, \tag{6.84}$$

where $\frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_l}$ can be solved from the GPDE (6.65).

The parameters of the output nonlinearity can be described by $\theta_{\mathcal{F}} = \begin{bmatrix} \theta_{\mathcal{F},1} & \theta_{\mathcal{F},2} \dots & \theta_{\mathcal{F},N_f} \end{bmatrix}$. Thus l^{th} component of the gradient with respect to the output nonlinearity becomes

$$\frac{\partial \delta(k)}{\partial \theta_{\mathcal{F},l}} = \frac{\partial \mathcal{F}\left(\mathbf{C}_p \mathbf{x}_p(k-d)\right)}{\partial \theta_{\mathcal{F},l}}, \tag{6.85}$$

where $\mathbf{x}_p$ does not depend on $\theta_{\mathcal{F},l}$.

The gradient of the input nonlinearity can be obtained similarly as the output nonlinearity by using the chain rule. If the parameters of the input nonlinearity are described by a vector $\theta_{\mathcal{G}} = \begin{bmatrix} \theta_{\mathcal{G},1} & \theta_{\mathcal{G},2} \dots & \theta_{\mathcal{G},N_g} \end{bmatrix}$, the l^{th} component of the gradient with respect to the input nonlinearity becomes

$$\frac{\partial \delta(k)}{\partial \theta_{\mathcal{G},l}} = \frac{\partial \mathcal{F}\left(\mathbf{C}_p \mathbf{x}_p(k-d)\right)}{\partial \mathbf{x}_p(k-d)} \frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_{\mathcal{G}_l}}, \tag{6.86}$$

where $\frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_{\mathcal{G}_l}}$ can be solved from the GPDE

$$\frac{\partial \mathbf{x}_p(k+1)}{\partial \theta_{\mathcal{G},l}} = \mathbf{\Phi}_p \frac{\partial \mathbf{x}_p(k)}{\partial \theta_{\mathcal{G},l}} + \mathbf{\Gamma}_p \frac{\partial \mathcal{G}\left(\mathbf{y}(k)\right)}{\partial \theta_{\mathcal{G},l}}, \tag{6.87}$$

where the initial values are $\frac{\partial \mathbf{x}_p(0)}{\partial \theta_{\mathcal{G},l}}$ can be assumed zero, because at $k = 0$ the output does not affect to the state variables.

6.6.3 Known output level saturation

When output level saturation is included in (6.83), HWDDBSFM can be written in analogous form to (6.5) as

$$\begin{cases} \mathbf{x}_p(k+1) = \mathbf{\Phi}_p \mathbf{x}_p(k) + \mathbf{\Gamma}_p \mathcal{G}\left(\mathbf{y}(k)\right) \\ \delta(k) = \text{SAT}\left(\mathbf{C}_p \mathbf{x}_p(k-d), \delta_{\max}\right) \end{cases}, \tag{6.88}$$

where $\delta_{\max}$ is the maximum allowable control signal, and the initial values are $\mathbf{x}_p(0) = \mathbf{x}_{p,0}$ as in (6.61). Due to the saturation function, the derivative of the human operator output δ is zero during saturation. Therefore, the derivative of the output becomes a piecewise function

$$\frac{\partial \delta(k)}{\partial \theta_l} = \begin{cases} \mathbf{C}_p \frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_l}, & \text{if } |\mathbf{C}_p \mathbf{x}_p(k-d)| < \delta_{\max} \\ 0, & \text{otherwise} \end{cases}, \tag{6.89}$$

which is applied in (6.63) when gradient-based minimization is used to find the optimal parameters of the DDBSFM.

6.6.4 Known output level and rate saturation

If both, output level and output rate saturation are included, the output of the HWDDBSFM is analogous to (6.12)

$$\delta(k) = \delta(k-1) + \mathrm{SAT}_R \left(\mathrm{SAT}_L \left(\mathbf{C}_p \mathbf{x}_p(k-d), \delta_{\max} \right) - \delta(k-1), d\delta_{\max} h \right), \tag{6.90}$$

where SAT_L is the output level saturation function and SAT_R is the output rate saturation function, and the initial values are $\mathbf{x}_p(0) = \mathbf{x}_{p,0}$ as in (6.61).

The derivative of $\delta(k)$ with respect to θ_l can now be written as follows.

$$\frac{\partial \delta(k)}{\partial \theta_l} = \begin{cases} \mathbf{C}_p \frac{\partial \mathbf{x}_p(k-d)}{\partial \theta_l}, & \text{if } \begin{cases} |\mathbf{C}_p \mathbf{x}_p(k-d)| < \delta_{\max} \\ |\mathbf{C}_p \mathbf{x}_p(k-d) - \delta(k-1)| < d\delta_{\max} h \end{cases} \\ 0, & \text{if } \begin{cases} |\mathbf{C}_p \mathbf{x}_p(k-d)| \geq \delta_{\max} \\ |\delta_{\max} - \delta(k-1)| < d\delta_{\max} h \end{cases} \\ \frac{\partial \delta(k-1)}{\partial \theta_l}, & \text{otherwise} \end{cases} \tag{6.91}$$

The identification can be performed by using the gradient-based minimization algorithms for (6.63) and the piecewise definition for the output derivative above.

6.7 Discussion

The human operator modeling problem is studied from the perspective of structural models. In this thesis, the structural model refers to the model, which is designed based on the model of the controlled system. The most significant advantages of such models are the possibility to retain the mapping from the machine parameters to the human operator model parameters, and the automatic solution to the model structure selection problem. Commonly in modeling and identification, the model structure selection plays a critical role. When using structural models for human operator, the structure is tied together in the model of the controlled machine, and the task at hand. Thus, the only problem in the modeling is to find a mapping from the measurements to the human control command by varying the free parameters of the model. It was shown how the mapping can be obtained in frequency and time domain.

The Modified Optimal Control Model (MOCM) is a well-known method for modeling the human operator in manual control tasks. The extension for the MOCM using the Hammerstein-Wiener structure was proposed. The Hammerstein-Wiener structure provides convenient tools

to account for nonlinearities in the output and input of the human operator model.

The challenge in the MOCM is that the mapping from the control weight and the Kalman filter intensity matrices are connected to the control gain and estimator matrices via Riccati equations. Thus, the direct gradient-based optimization is difficult, because the solution of the Riccati equations should be guaranteed. Using the return difference condition the mapping from the weight and intensity matrices can be expressed as [8]

$$\begin{aligned} & f + \left((i\omega \mathbf{I} - \mathbf{A}_0)^{-1} \mathbf{B}_0 \right)^H \mathbf{Q}_0 \left(i\omega \mathbf{I} - \mathbf{A}_0 \right)^{-1} \mathbf{B}_0 \\ = & \left(1 + \tilde{\mathbf{g}}_p \left(i\omega \mathbf{I} - \mathbf{A}_0 \right)^{-1} \mathbf{B}_0 \right)^H f \left(1 + \tilde{\mathbf{g}}_p \left(i\omega \mathbf{I} - \mathbf{A}_0 \right)^{-1} \mathbf{B}_0 \right). \end{aligned} \tag{6.92}$$

The control weighting $\mathbf{Q}_0$ (and the estimator noise intensity $\mathbf{W}_1$) can then be optimized using the frequency domain approach. Alternatively, one could use the LQR necessary conditions of optimality, which leads to the differential equation system

$$\begin{bmatrix} \dot{\boldsymbol{\chi}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_0 & -\mathbf{B}_0 f^{-1} \mathbf{B}_0^T \\ -\mathbf{Q}_0 & \mathbf{A}_0^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\chi}(t) \\ \mathbf{p}(t) \end{bmatrix}, \tag{6.93}$$

where $\mathbf{p}$ denote the costates of $\boldsymbol{\chi}$. This approach was proposed in [111] but not implemented. The implementation requires the estimation of the costates with unknown initial value, and fixed end value. Thus it becomes a two-point boundary value problem. Moreover, the adaptive LQG approach proposed in [53] could be utilized to solve the parameter estimation problem and simultaneously preserve the optimality conditions.

In this thesis, a different approach was taken. The model structure is chosen to be equal to the MOCM, but the optimality assumptions are relaxed. This enables direct gradient-based optimization using the control and estimation gains rather than the weigting matrices.

The proposed frequency domain parameter estimation method can be used to match especially the frequency domain characteristics of the model to the measured frequency domain characteristics. This is important in many human-machine systems. The introduction of the return difference condition constraints guarantee the existence of the inverse optimal solution with $\mathbf{Q}_0 \geq 0$ and $\mathbf{N}_0 = 0$. The LMI-based inverse optimal solutions are presented for the controller and the estimator matrices.

In addition, the time domain gradient-based identification method is proposed. The identification is based on iterative solutions of the gradient propagation difference equations (GPDEs). Moreover, the Hammerstein-Wiener extension of the DDBSFM is presented, and the corresponding gradients for the parameter estimation are given. In the parameter estimation, the known and unknown parametric nonlinearities can be accounted for. The existence of the inverse optimal solutions cannot be guaranteed for $\mathbf{Q}_0 \geq 0$ and $\mathbf{N}_0 = 0$, but the LMI-based solutions for the optimal controller (and the estimator) where $\mathbf{Q}_0 \geq 0$ and $\mathbf{N}_0 \neq 0$ are presented.

The presented methods are suitable for batch estimation of the parameters. This is enough in cases where the human repeatedly performs a task (or a small set of tasks, which can be recognized and classified), which has a beginning and the end. Thus, the parameter estimation can be done using the data from several executions of the task. However, the tracking type of tasks would require continuous estimation. The simultaneous estimation of the human operator

model states $\mathbf{x}_p(t)$ and the parameters $\theta(t)$ is a nonlinear estimation problem, which can be treated using conventional nonlinear estimation methods.

Additional challenge for online implementation is the estimation of the time delay. The theory of hereditary systems can be used to identify the time-varying system parameters time delay as described in [83]. The simultaneous estimation of the time delay and the linear model parameters is discussed in [23], where a recursive delay identifier using the Extended Kalman Filter was proposed. Nevertheless, the presented structures of the DBSFM or the DDBSFM have more unknown parameters than states. Thus, in cases with low number of input signals, the online estimation of all parameters is not necessarily possible. Instead, the control gain $\mathbf{g}_p$ could be fixed and the neuromotor time constant τ_n, time delay τ_p and the estimator gain $\mathbf{F}$ can be set as free parameters.

In this thesis, the discussion is restricted to feedback-type of human operator models. However, as it is intuitively clear, and as important works such as [76] proves, the human operator is a feedback-feedforward controller. As the human operator is learning, the strenght of the feedforward path increases. In the future, in order to obtain human models which are more realistic, a model which could be made adaptive to the changes in the operating point and which would include the feedforward structure is needed.

Yet another point which should be mentioned is the model order. The MOCM as well as the data-based versions described in this thesis result in high order models. The model reduction methods for MOCM were studied in [34, 35], which could be utilized here as well. In addition, the model of the controlled plant should be of minimal order. Otherwise, the overlearning might occur in the parameter estimation.

7 Identification of human model from experimental data

The human operator modeling methods from the perspective of structural models were studied in the previous chapter. The aim of the chapter was to develop modeling methods which would describe the continuum of human operator models with respect to the operating point of the controlled machine. As described in Section 6.1, such property would be desirable when modeling human operators in machine work. In addition, the aim of the previous chapter was to develop modeling methods whose parameters can be identified using incomplete measurement data from normal work. This property is also favorable from the machine work perspective, since in typical boom or trolley cranes the exact states of the joint angles or the rope angles are not available as measurements. The third objective of the previous chapter was to develop models which would exploit the information about the known nonlinearities present in the HMI (and possibly in the measurements) of the system. Such nonlinearities might include, for example, saturation of the control signal level or rate. This is also desirable property from the machine work perspective, since based on author's experience, the real-world cranes and machines typically include the control level saturation and possibly ramps (rate limiters) to prevent fast changes in the control signal. The modeling objectives can be summarized as follows:

1. The model should retain the mapping from the parameters of the controlled machine so that the adaptivity to the varying operating points could be realized.

2. The model should perform as well as the widely accepted simple linear models such as ARX.

3. The parameters of the model should be efficient to estimate from incomplete normal task execution data.

4. The model should be able to account for the possible strong nonlinearities in the HMI.

There exists several models available for modeling the human operator such as simple ARX, neural networks, fuzzy models, etc. The aim of this chapter is to implement the models proposed in Chapter 6 to model a human operator in a point-to-point task of a trolley crane simulator. In particular, the aim is to find out whether the proposed modeling methods are able to predict the behavior of the human operator as well as simple conventional model structures as ARX or Hammerstein-Wiener polynomial models. Moreover, the aim is to study whether the Hammerstein-Wiener model structure is favorable in cases where the HMI contains strong nonlinearities such as saturations or rate limiters.

In this chapter, the human operator models are identified using the proposed new methods as well as the conventional ones for the linear HMI case (no significant saturation or rate limit) and the nonlinear HMI case (heavy saturation and rate limit in the control signal). A numerical example of the inverse optimal control solution for DBSFM is presented and the corresponding return difference conditions are shown in the frequency domain. In addition, the numerical suboptimal inverse solution for DDBSFM is presented.

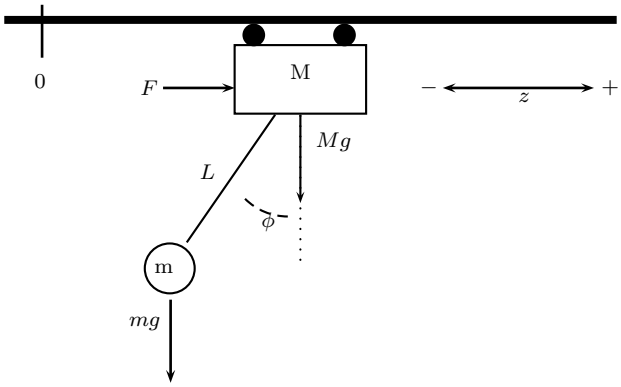


Figure 7.1: A free-body diagram of the trolley crane system.

7.1 Experimental setup

In order to compare the human operator identification and modeling methods a trolley crane simulator was used. The simulator was developed by the author to test the concept of human skill adaptive control [154]. In the simulator, the dynamics of the trolley crane system is simulated in Simulink and the visualization is done with Matlab.

The structure of the trolley crane system is shown in Figure 7.1. A load with weight m is connected to a trolley with weight M via a rope. The rope is assumed to be stiff and its mass is assumed to be zero. The trolley can be moved in a horizontal direction by introducing the control force F. The trolley is assumed to be affected by linear friction force proportional to the velocity of the trolley. The frictional coefficient is b. The position in the horizontal direction is denoted by z, the length of the rope by L, and the rope angle by ϕ.

The mathematical model of the trolley crane system can be derived based on the Newtonian mechanics by summing the force components in horizontal and vertical directions. The equilibrium equations are as follows [107].

$$\begin{cases} (M+m)\ddot{z} + b\dot{z} + mL\cos(\phi)\ddot{\phi} + mL\sin(\phi)\dot{\phi}^2 = F \\ mL\cos(\phi)\ddot{z} + mL^2\ddot{\phi} + mgL\sin(\phi) = 0 \end{cases} \tag{7.1}$$

The simulator exploits the nonlinear equations of the trolley crane. In order to obtain the parameters of the MOCM, one needs to linearize the model around an operating point. The linearized model of the trolley crane is

$$\begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -b & \frac{gm}{M} & \\ 0 & 0 & 0 & 1 \\ 0 & \frac{b}{L} & -\frac{g(M+m)}{ML} & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{ML} \end{bmatrix} u \\ \\ \mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} \end{cases}, \tag{7.2}$$

Table 7.1: Physical parameters of the trolley crane model.

Description	Symbol	Value
Load mass	m	2.00 kg
Trolley mass	M	1.00 kg
Rope length	L	1.00 m
Trolley friction coefficient	b	1.00 $\frac{\text{Ns}}{\text{m}}$

where the state vector is defined as $\mathbf{x} = \begin{bmatrix} z & \dot{z} & \phi & \dot{\phi} \end{bmatrix}^T$ and the input u is the control force force F. It is important to note that only the output z is assumed to be measurable, because the angle measurement is not usually available in real-life industrial cranes.[16] The physical parameters of the trolley crane used in the simulator experiment are shown in Table 7.1

7.2 Experimental data

In the experiment, a human subject performed a PTP task with the trolley crane simulator. In the task, the objective was to transfer the cart from initial point (-0.75 m) to terminal point (0.75 m) and then dampen the swinging of the load. The initial values for $\dot{z}$, ϕ and $\dot{\phi}$ were zero. The task was defined to be finished when the cart position close enough to the terminal point (a predefined threshold), and the values of cart speed, rope angle, and angular velocity were under predefined thresholds. During the experiment, the values of z, $\dot{z}$, ϕ, and $\dot{\phi}$ were recorded into a file. In addition, the values of the operator's control signal δ were recorded into the same file. In the identification, only the signals of δ and z were used (see the previous footnote[16]).

The sampling time used in the simulator was $h = 0.065$ s. When using the Bogacki-Shampine differential equation solver and a fixed step with 0.065 s, the simulator time corresponds to the real world time well enough with the computer used to run the simulator (Intel(R) Core(TM)2 CPU T7200@2.00GHz, RAM 3.00 GB).

The starting time of the task was randomized so that the human operator could not see the simulation visualization until a random time interval was passed. In this way the human operator delay from perception to action can be estimated accurately. For simplicity, the rope length was kept constant during the task execution. For each experiment, the human operator performed the task 150 times, having unlimited rest time after each 10 trials. Every second of

[16] In principle, this is not a good assumption, because it is intuitively clear that the human operator will exploit the information about the rope angle. However, because (even today) the real-life cranes do not include the required sensors (cameras, ultrasound measurements, etc.) to measure the rope angle, it is reasonable to omit this measurement in the modeling, even though it is available in the simulator. Moreover, even if the *measurement* is not used, it is shown in the MOCM, DBSFM and DDBSFM control weighting matrices in Sections 7.3.1 and 7.3.2 that the human operator actually uses the *estimate* of the angle to determine the conrol law.

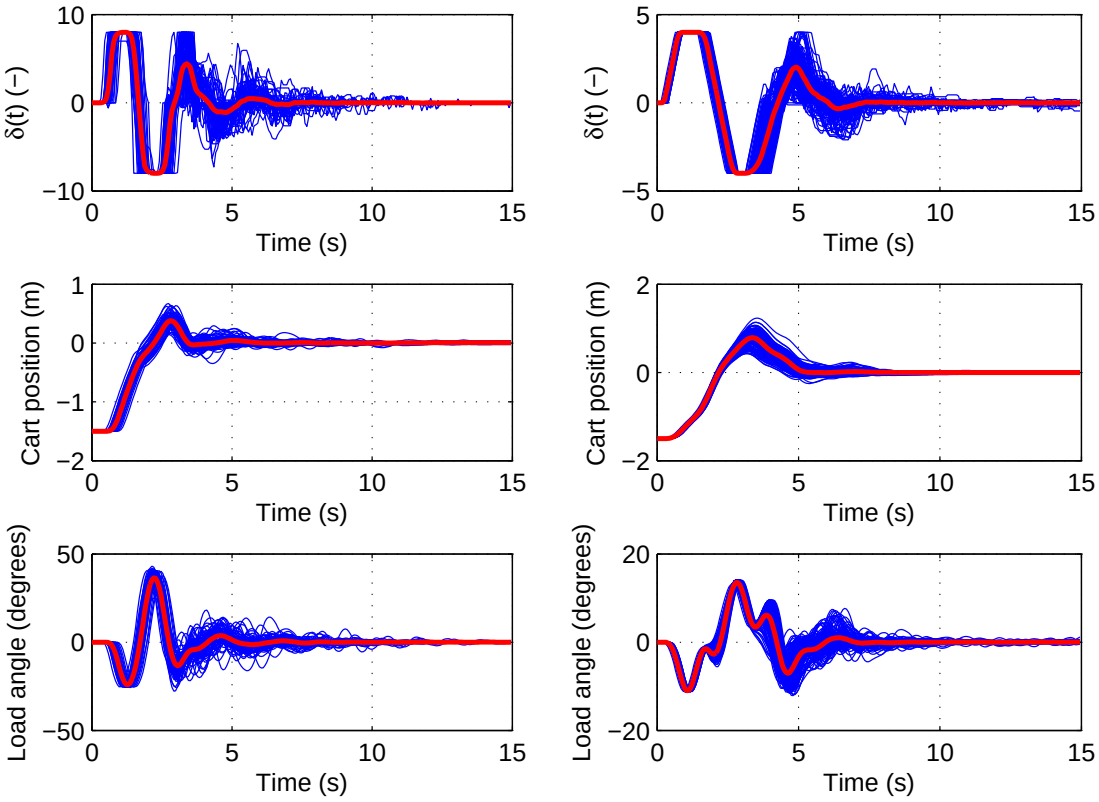


Figure 7.2: Time domain data when there are no nonlinearities in the HMI (left) and in the presence of level and rate saturation of the control signal (right). The control signal given by the operator (uppermost), the cart position (middle) and the load angle (bottom) for the training data. Individual responses are given as blue shallow lines and the average over the repetitions as red thick line.

the trials was chosen for training data, and the rest for validation data. Thus, the training and the validation data sets consisted of 75 trials of the task. The data from the training and the validation sets were averaged. The effect of learning was not considered because the human operator had practiced the task over 100 times before the data were recorded.

The human response as well as the response of the controlled system can be seen in Figure 7.2, where the training data set is presented for linear (left) and nonlinear (right) HMIs. Usually, in order to reduce the effects of high frequency noise peaks caused by unintended movements of the control levers rate limiters (ramps) are used to filter the control signal before introducing it into the control system. The effect of ramps needs to be accounted for in the identification process. In this thesis it is done by using the Hammerstein-Wiener structured models. Therefore, a separate test data set was gathered for that purpose.

In the linear case of Figure 7.2 the the sensitivity of the control interface is set so high ($\delta_{\max} = 8$)

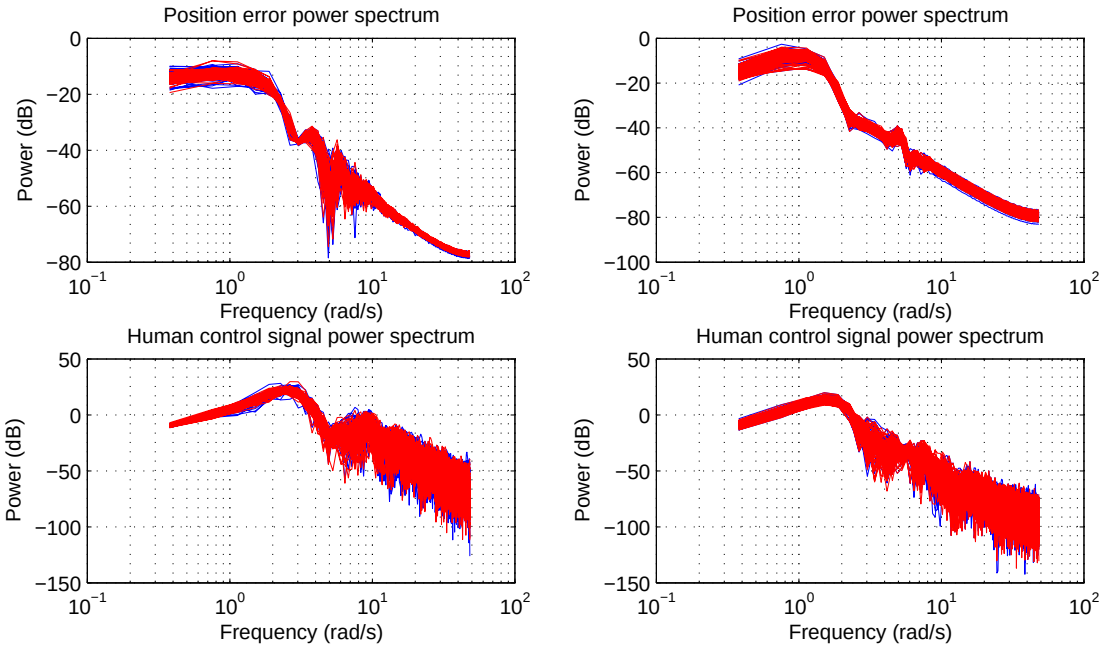


Figure 7.3: Frequency domain data when there are no nonlinearities (left) in the HMI and with input level and rate saturation (right). The power spectrum of the position error signal and the human control signal for the training data (blue) and the validation data (red). Bot data sets contain 75 experimental frequency responses.

that the human operator use full controls more rarely. The output of the joystick the human operator uses to control the cart is in the range $[-1, 1]$. In the linear case, the gain of the joystick was set to eight, which is now the maximum absolute value of the control signal. No rate limiter was used in the linear case ($d\delta_{\max} = \infty$).

Although the objective was that the control signal would not saturate in the linear case, in practice the operator still used full controls at times. In the nonlinear control interface the human control signal is passed through level and rate saturation elements. From the responses one can get an impression of the stochastic nature of the human task execution. The sample average of the signals over all repetitions is presented by thick red line. From the control signal one can see that the first two large control actions, maximum positive and then maximum negative are rather consistent over trial, although the delay varies slightly. However, the variation of the control signal is significantly greater for small movements from 3 s forward (in the linear case), that is, the phase when the human operator is trying to compensate the swinging of the load. The smaller the swinging angle becomes, the harder it is for the human operator to estimate the angular velocity and thus introduce correct control actions. In addition, the human operation in high frequencies has significantly greater variance than in low frequencies, which can also be seen from Figure 7.3.

The power spectrum of the data set gathered using the linear HMI is shown in Figure 7.3. The figure shows that the power of the input data decreases drastically after the frequency 2 rad/s. From the identification point of view the resulting estimator does not necessarily describe the human operator's dynamics well at high frequencies with low power in the training data. However, if such a transfer function is found which models the frequency response of the human operator well, the resulting estimator can be used to describe the human operator's dynamics for similar inputs as the training data.

In the nonlinear case, the control gain was set to four ($\delta_{\max} = 4$), which limits the control signal in the range $[-4, 4]$. In addition, the maximum change allowed for the control signal in a time step was set to eight ($d\delta_{\max}h = 8$). The power spectra for the data gathered from the nonlinear case are shown in Figure 7.3.

7.3 Identification results

7.3.1 Results for linear HMI

In the first experiment, the parameters of the models described in Chapter 6 were estimated by using the data shown on the left in Figure 7.2. The time domain estimation was performed using the ensemble average of the input and output signals. The frequency domain estimation was performed based on the ensemble average of the individual frequency responses of the trials.

The parameters of the MOCM and HWMOCM models were determined using the sequential quadratic programming algorithm implemented in Matlab Optimization Toolbox [13] "fmincon" routine with constraints for the positive definitess of $\mathbf{Q}_0$ and $\mathbf{W}$. The routine allows to use linear and nonlinear inequality and equality constraints. The unknown parameters were the weights $\mathbf{Q}_0(3,1)$ and $\mathbf{Q}_0(3,3)$, the noise intensities $\mathbf{W}(1,[1,2,3,4])$, $\mathbf{W}(2,[2,3,4])$, $\mathbf{W}(3,[3,4])$, $\mathbf{W}(4,4)$, $\mathbf{V}_y$ and $\mathbf{V}_u$, as well as τ and τ_n. Thus, in total, 16 parameters were to be estimated. The initial values were chosen so that $\mathbf{W}$ is an identity matrix, $\mathbf{Q}_0(3,3) = 1$, $\mathbf{Q}_0(1,3) = 0$, $\tau_n = 0.1$ s, and $\tau = 0.45$ s. The control weight for the position was kept as constant ($\mathbf{Q}_0(1,1) = 1$). The resulting model parameters are presented in Appendix D.1 for MOCM and in Appendix D.2 for HWMOCM.

The parameters of DBSFM, DDBSFM and HWDDBSFM were determined using the gradient-based sequential quadratic programming algorithm implemented in Matlab Optimization Toolbox routine "fmincon". The return difference criteria were implemented so that the problems (6.35) and (6.36) were solved by using the gradient-based sequential quadratic programming method. However, because the return difference conditions might have several minima, the optimization was ran from three different initial values for ω.

The unknown parameters for DBSFM were $\mathbf{g}_p$, $\mathbf{F}([1,2,\ldots,7],1)$, τ_n, and τ. For DDBSFM and HWDDBSFM the delay should be determined beforehand, thus, the number of free parameters for DBSFM was 15 and for DDBSFM and HWDDBSFM 14. The initial values for the DBSFM were set as described in Section 6.4.3. The initial values for the required parameters are $\tau_n = 0.1$ s, $\tau = 0.4$ s, $\mathbf{Q}_0 = \mathrm{diag}(1,0,1,0,0,0,0)$, and $\mathbf{W}_1 = \mathrm{diag}(1,1,1,1,0,0,100)$. The initial values for DDBSFM and HWDDBSFM were set equal to the ones for DBSFM except for $a_n = 0.3517$ (corresponds to $\tau_n = 0.15$ s) and $\tau = 0.39$ s, and thus the dead time lag is 6

Table 7.2: Performance comparison and the numbers of free parameters of models for linear HMI data.

Model	$V(\theta)$	$V_{\text{pos}}(\theta)$	$V_{\text{angle}}(\theta)$	N_θ
MOCM	20.5155	0.2147	0.0573	16
DBSFM	8.0531	0.1615	0.0354	15
DDBSFM	6.0969	0.1208	0.0487	14
ARX898	12.8882	0.0822	0.0634	17
HWOE889	5.2743	0.0484	0.0417	16
HWMOCM	13.7220	0.1669	0.0694	16
HWDDBSFM	5.4670	0.1860	0.0447	14

samples.

In addition to the structural models, the conventional time series models were identified for comparison. The identification was performed as described in the author's publication [157], where the model orders and structures were studied for the same data. In particular, the publication focused in time series models. The best results were achieved using the ARX and Hammerstein-Wiener Output Error Model (HWOE) structures. The latter provided the best results as it can account for the saturation in the control signal. In this thesis, the order for the structural models was set to eight, because it is the order of the MOCM, DBSFM and DDBSFM for the system at hand. In [157], the corresponding number of free parameters in the numerator were 9 for ARX and 8 for HWOE. The dead time delay lags for the models were 8 for ARX and 9 for HWOE. Thus, the number of free parameters in total is now 17 for ARX and 16 for HWOE. In the following, the corresponding models are referred as ARX898 and HWOE889.

The training of the HWOE model was performed by using the prediction error method with known saturation nonlinearity. In [157] it was concluded that the linear time series models trained using the regular prediction error method and data from execution of a single task provide poor simulation performance. In practice, one should use the frequency weighting to emphasize the frequencies with high power in the training data, if the training data is poor from the system identification perspective. As explained previously, the power of the experimental data used in the experiment is high only in a small range of frequencies.

The time domain results for the linear HMI are shown in Figure 7.4. On the left, the control signal given by the linear versions of MOCM (numerical values of the parameters are described in (D.1)–(D.7)), DBSFM and DDBSFM are shown. On the right, the results given by the model HWMOCM (numerical values of the parameters are described in (D.8)–(D.14)) and HWDDBSFM are shown.

The control signal is obtained by setting the identified model to execute the same task as the operator with the same nonlinear simulator (including the same saturation limit). The responses given by the Hammerstein-Wiener models seem to be better than the responses given by the linear ones.

Numerical comparisons of the identification results in Table 7.2 confirm the visual inspection

of Figure 7.4. The performance index $V(\theta)$ is defined as (6.14), where δ_m is the value of the ensemble average of the human control signal based on the validation data, and δ is the control signal given by the model when the model is set to execute the same task as the human operator by using the same simulator. However, one should note that the performance values $V(\theta)$ are comparable to the results reported in [157] only if multiplied by two. The performance metrics $V_{\text{pos}}(\theta)$ and $V_{\text{angle}}(\theta)$ are similar to $V(\theta)$, but the error between the measured and simulated control signal is replaced with the error between the measured and simulated position and the angle signals, respectively.

According to the numerical results with respect to $V(\theta)$, the best performance is given by HWOE889, HWDDBSFM and DDBSFM. Comparison of the values of $V_{\text{pos}}(\theta)$ suggests that the simulated trajectory of the trolley crane given by the polynomial models is closer to the measured trajectory than the responses given by the structural models. On the contrary, the performance of DBSFM is best with respect to $V_{\text{angle}}(\theta)$. Note that none of the models used the angle control as input. The differences between the performances of the structural models and the polynomial models with respect to the position trajectory could be explained by the fact that the structural models assume the internal model of the human operator to match with the true dynamics of the controlled plant. However, if that is not the case and, especially if the true plant is nonlinear (as it is in this experiment), the structural models might result in good performance with respect to the control signal but not with respect to the response of the system.

In addition to the performance indices, the number of free parameters of the models are shown in Table 7.2. In MOCM models there are 16 parameters, whereas in DBSFM and DDBSFM models the number of free parameters is 15 and 14, respectively.

The frequency responses given by the identified models are shown in Figure 7.5. The Bode diagrams on the left describe the frequency responses given by the linear models and on the right the linear cores of the Hammerstein-Wiener models. Note that only DBSFM and ARX898 are identified by using the frequency domain approach. From the responses one can note that MOCM trained by using the simulation approach described in Section 6.3.3 does not result in a zero dB static gain. In general, the time series models capture the frequency domain characteristics well, because they are more flexible than the structural models. However, ARX898 fails to capture the phase curve at high frequencies.

7.3.2 Inverse optimal control and estimation solutions

7.3.2.1 Data-based state feedback model

The 0-plant (B.8) and the feedback gain (B.9) for controlling the experimental trolley crane simulator obtained by the identification of the DBSFM are

$$\mathbf{B}_0^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{\mathbf{g}}_p = \begin{bmatrix} 170.850141 & 135.484312 & 155.731313 & 99.904240 & 2.932618 \\ & & & 39.207127 & 9.040173 \end{bmatrix}$$

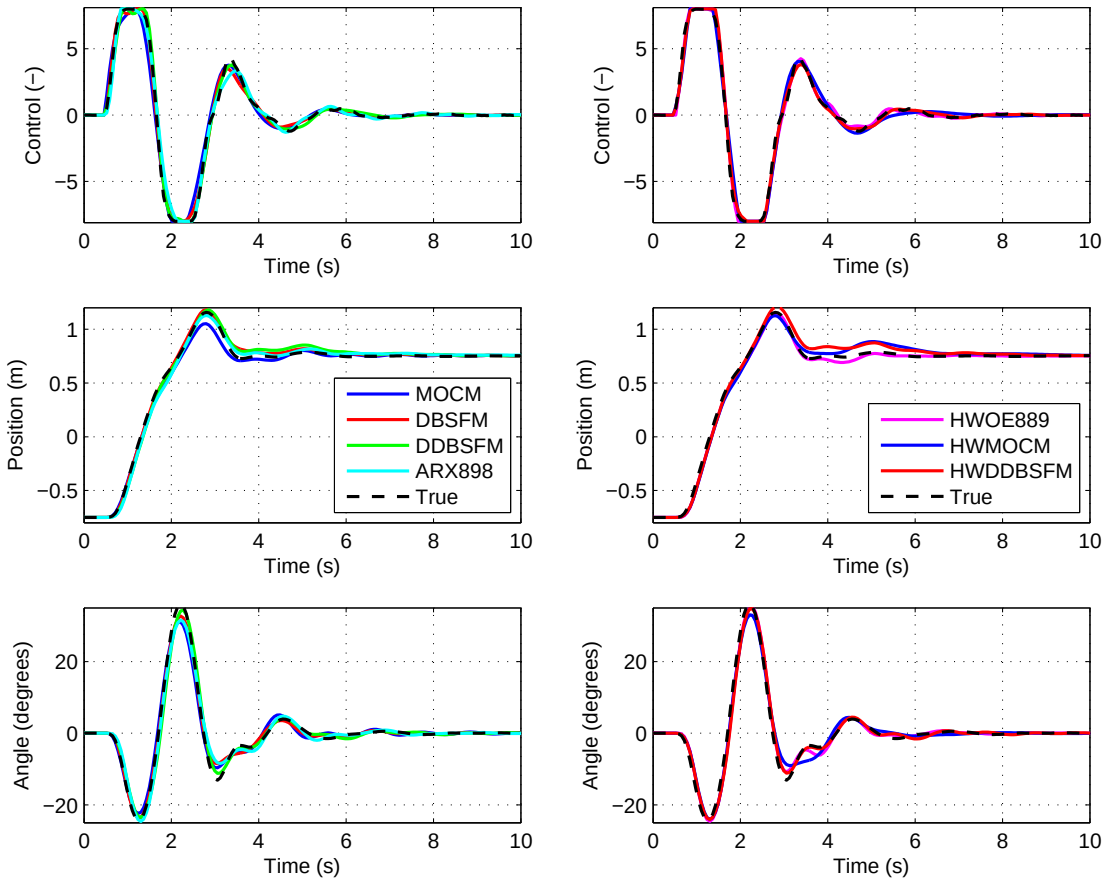


Figure 7.4: The time domain responses given by the identified models when the model is set to execute the operator's task in the nonlinear simulator. The black dashed line represents the ensemble average of the measured signal based on the validation data.

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 19.620000 & 0 & -23.641185 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -29.430000 & 0 & 23.641185 & 0 & -1 \\ 0 & 0 & 0 & 0 & -11.820593 & -1.970099 & 1 \\ 0 & 0 & 0 & 0 & 23.641185 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The last element in $\tilde{\mathbf{g}}_p$ is $\frac{1}{\tau_n}$, where $\tau_n = 0.110617$ s obtained also in the identification process.

Before solving the inverse optimal control problem, the return difference conditions should be studied. The return difference conditions (6.31) and (6.33) are plotted as functions of the frequency in Figure 7.6. The curves are zoomed around the mimima values of the conditions. One can see that the controller and the estimator both fulfill the return difference condition. Thus, it is possible to solve the inverse optimal control problem with $\mathbf{N}_0 = 0$, $\mathbf{Q}_0 \geq 0$, $\mathbf{N}_1 = 0$

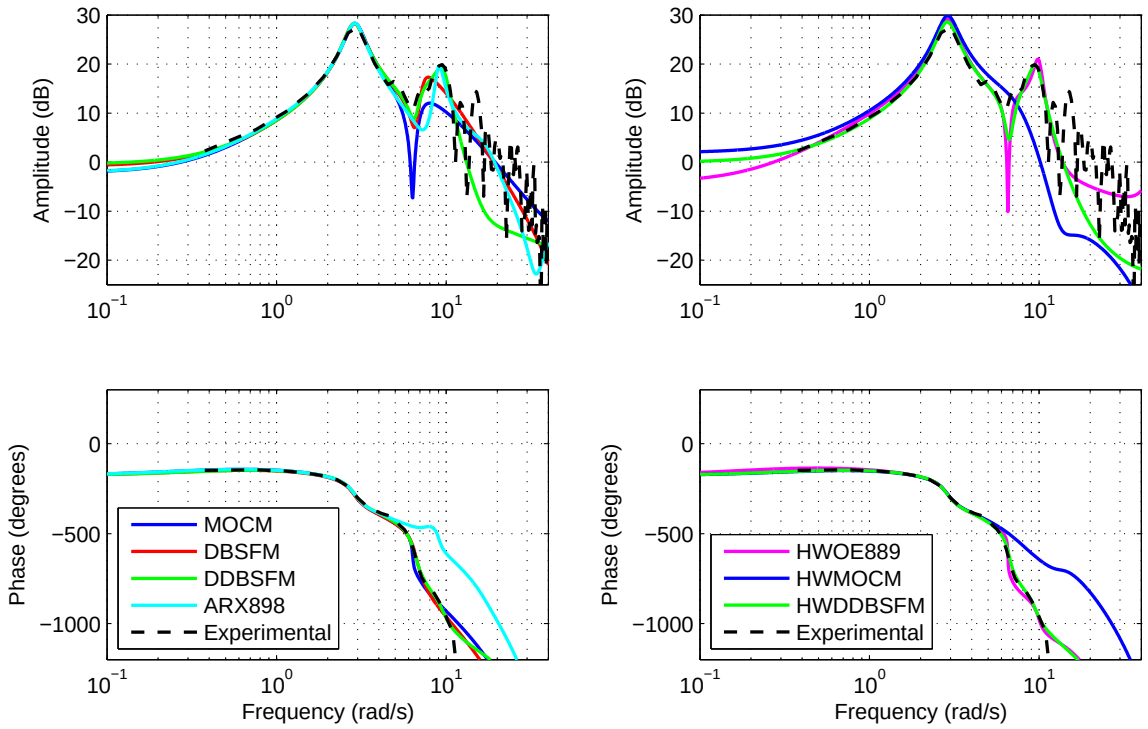


Figure 7.5: Comparison of the frequency responses given by the identified models. The experimental frequency response for the validation data is obtained by using the periodograms method [95].

and $\mathbf{W}_1 \geq 0$.

By using the 0-plant and the identified feedback gain, the inverse optimal control problem (6.42) was solved. The LMI problem was solved using the YALMIP interface [94] and the SeDuMi [91] solver both developed for Matlab. The resulting unknown parameters are shown in (D.15)–(D.18) which, by using by (6.40) produce the feedback gain

$$\hat{\tilde{\mathbf{g}}}_p = \begin{bmatrix} 170.845617 & 135.468052 & 155.725503 & 99.915351 & 2.918941 \\ & & & 39.201378 & 9.048717 \end{bmatrix}$$

that is close to the gain $\tilde{\mathbf{g}}_p$ estimated by the DBSFM identification. For the accuracy required from the solution of the inverse optimal control problem, the exact values will produce the gain matrix that has the norm $|\tilde{\mathbf{g}}_p - \hat{\tilde{\mathbf{g}}}_p| \approx 1 \cdot 10^{-5}$. Because the state feedback controller fulfills the return difference condition (6.31), the accuracy can be improved by tightening the solution tolerance for the LMI solver.

The inverse solution for optimal estimation parameters was tested using the DBSFM model

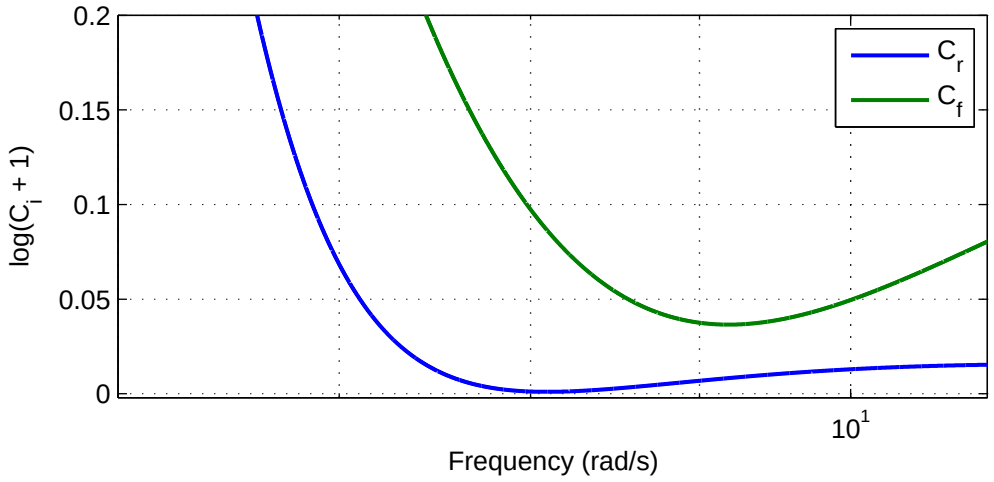


Figure 7.6: The return difference condition constraints (6.31) and (6.33) with frequency for the identified DBSFM.

identified previously. The 1-plant (B.16) and the estimator gain $\mathbf{F}$ for the estimated model are

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 19.620000 & 0 & -23.641185 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -29.430000 & 0 & 23.641185 & 0 & -1 \\ 0 & 0 & 0 & 0 & -11.820593 & -1.970099 & 1 \\ 0 & 0 & 0 & 0 & 23.641185 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.040173 \end{bmatrix}$$

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{F}^T = \begin{bmatrix} 13.304589 & -100.000000 & -3.094244 & 87.011085 & -72.515887 \\ & & & 82.898872 & 365.798422 \end{bmatrix}.$$

The inverse optimal estimation problem (6.46) was solved using the YALMIP interface and the SeDuMi solver. The resulting unknown parameters are shown in (D.19)–(D.22). By using the solved parameters (with the accuracy in the text) the estimator gain matrix by (6.44) becomes

$$\hat{\mathbf{F}}^T = \begin{bmatrix} 13.305078 & -100.000000 & -3.095654 & 87.010721 & -72.517244 \\ & & & 82.901188 & 365.802377 \end{bmatrix}.$$

If the accurate values are used, the difference between the true estimator gain matrix and the one given by the inverse solution is $|\mathbf{F} - \hat{\mathbf{F}}| \approx 4.5 \cdot 10^{-6}$. Again, because also the estimator fulfills the return difference condition, by tightening the solution accuracy requirement for the LMI solver, the more accurate inverse solution for the estimator can be found.

7.3.2.2 Discrete data-based state feedback model

For the DDBSFM, no constraints guaranteeing the existence of the inverse solution were introduced. Therefore, only the suboptimal inverse solutions with nonzero crossterms can be found.

The 0-plant (6.47) and the feedback gain $\tilde{\mathbf{g}}_p$ for controlling the experimental trolley crane simulator obtained by the identification of the DDBSFM are

$$\boldsymbol{\Phi}_0 = \begin{bmatrix} 1 & 0.062947 & 0.040145 & 0.000878 & -0.045165 & 0.002759 & 0.000977 \\ 0 & 0.937931 & 1.209170 & 0.040145 & -1.130053 & 0.115808 & 0.016903 \\ 0 & 0.002046 & 0.939343 & 0.063676 & 0.044984 & -0.002753 & -0.000972 \\ 0 & 0.061629 & -1.833828 & 0.939343 & 1.119520 & -0.115344 & -0.016645 \\ 0 & 0 & 0 & 0 & 0.282363 & -0.099690 & 0.038879 \\ 0 & 0 & 0 & 0 & 1.196283 & 0.880504 & 0.046603 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\Gamma}_0^T = \begin{bmatrix} 0.000002 & 0.000064 & -0.000002 & -0.000063 & 0.000098 & 0.000071 & 1 \end{bmatrix}$$

$$\tilde{\mathbf{g}}_p = \begin{bmatrix} 6.136411 & 4.572721 & 4.448096 & 3.277818 & -0.031885 \\ & & & 1.449804 & 0.280443 \end{bmatrix}.$$

The last element in $\tilde{\mathbf{g}}_p$ is $a_n \approx 0.28$ which describes the neuromotor time constant in discrete time domain.

By using the 0-plant and the identified feedback gain, the discrete inverse optimal control problem (6.78) was solved using the YALMIP interface and the SeDuMi solver similarly as in the continuous case. The resulting unknown parameters are shown in (D.23)–(D.26) which produce the feedback gain from (6.76)

$$\hat{\tilde{\mathbf{g}}}_p = \begin{bmatrix} 6.136227 & 4.572586 & 4.447955 & 3.277733 & -0.031894 \\ & & & 1.449770 & 0.280454 \end{bmatrix}$$

This is close to the gain $\tilde{\mathbf{g}}_p$ estimated by the DBSFM identification. For the accuracy required from the solution of the inverse optimal control problem, the exact values will produce the gain matrix that has the norm $|\tilde{\mathbf{g}}_p - \hat{\tilde{\mathbf{g}}}_p| \approx 1.94 \cdot 10^{-6}$.

The inverse solution for optimal estimation parameters was tested using the DBSFM model identified previously. The 1-plant (B.16) and the estimator gain $\mathbf{F}$ for the estimated model are

$$\boldsymbol{\Phi}_1 = \begin{bmatrix} 1 & 0.062947 & 0.040145 & 0.000878 & -0.045165 & 0.002759 & 0.000821 \\ 0 & 0.937931 & 1.209170 & 0.040145 & -1.130053 & 0.115808 & 0.011430 \\ 0 & 0.002046 & 0.939343 & 0.063676 & 0.044984 & -0.002753 & -0.000817 \\ 0 & 0.061629 & -1.833828 & 0.939343 & 1.119520 & -0.115344 & -0.011201 \\ 0 & 0 & 0 & 0 & 0.282363 & -0.099690 & 0.030204 \\ 0 & 0 & 0 & 0 & 1.196283 & 0.880504 & 0.040039 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.719557 \end{bmatrix}$$

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

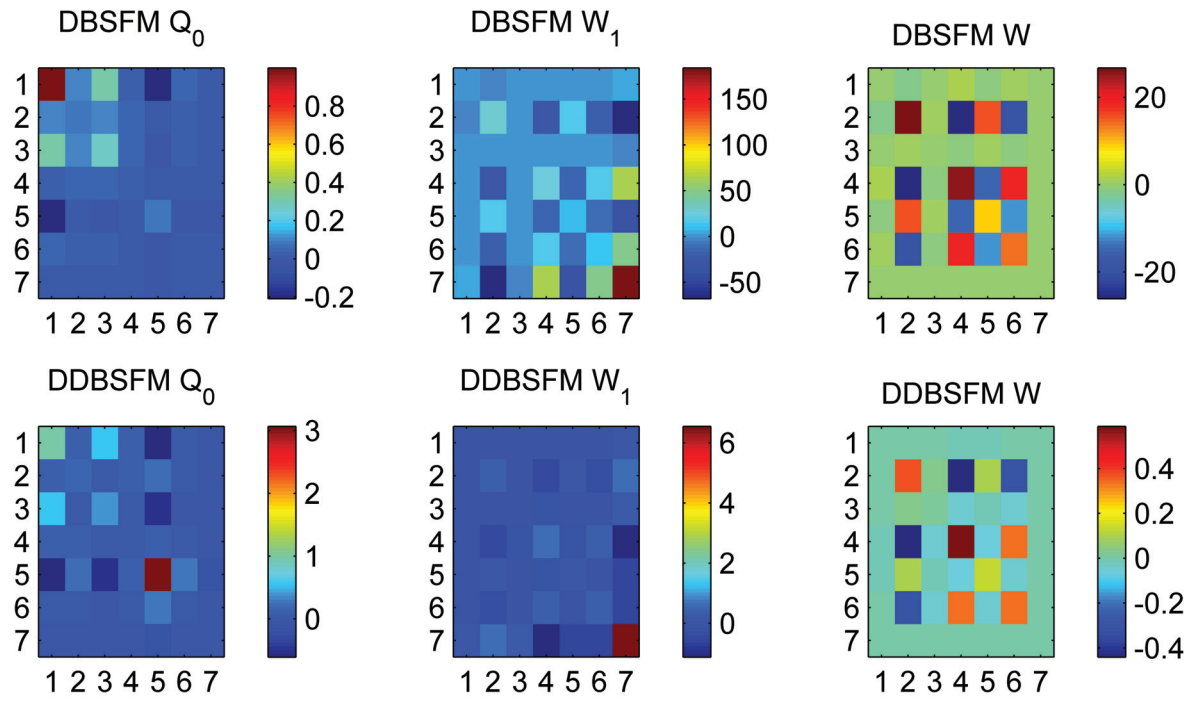


Figure 7.7: Visualization of the control weighting $\mathbf{Q}_0$ and noise intensity matrices $\mathbf{W}_1$ for DBSFM (upper) and DDBSFM (lower) given by the inverse solutions. The matrix $\mathbf{W}$ is equal to $\mathbf{W}_1$ except the fifth row and column are set to zero.

$$\mathbf{F}^T = \begin{bmatrix} 0.321439 & 1.636567 & 0.408693 & -2.861647 & -1.006822 \\ & & & -1.031625 & 10.776567 \end{bmatrix}.$$

The inverse optimal estimation problem (6.82) was solved using the YALMIP interface and the SeDuMi solver. The resulting unknown parameters are shown in (D.27)–(D.30). By using the solved parameters (with the accuracy in the text) the estimator gain matrix by (6.80) becomes

$$\hat{\mathbf{F}}^T = \begin{bmatrix} 0.321442 & 1.636543 & 0.408694 & -2.861601 & -1.006815 \\ & & & -1.031601 & 10.776536 \end{bmatrix}.$$

Using the exact numerical values, the norm between the true and estimated gain matrix becomes $|\mathbf{F} - \hat{\mathbf{F}}| \approx 5.7 \cdot 10^{-8}$.

Visualization of the control weigthing and estimator noise intensity matrices for DBSFM and DDBSFM given by the inverse solutions are shown in Figure 7.7. The control weighting matrices describe the objective of the human operator in the control task. In addition to the matrix $\mathbf{Q}_0$, the cross-term weighting $\mathbf{N}_0$ also describes the objective of the human operator. However, since the DBSFM fulfills the inverse optimality criterion, the crossterm is zero. For the DDBSFM the crossterm is nonzero. The noise intensity matrices $\mathbf{W}_1$ describe the level of uncertainty

of the human operator model in different states. Although the control and estimator gains depend on the initial values of the identification algorithm, it can be seen that DBSFM and DDBSFM both result in similar solutions.

The DBSFM emphasizes the position control weight in performance index (6.39). In addition, there crossterm between the first (position) and the third (angle) state is nonzero. For the estimator, the greatest uncertainty (of the actual physical states of the system) is in the second (cart speed) and the fourth (angular velocity) states, as well as in the delayed states. This is intuitively reasonable, since it is signigicantly more difficult for the human operator to estimate the derivatives than the actual values. The greatest difference between the DBSFM and DDBSFM is in the control weighting matrices, in particular, the first delayed state (the fifth state). However, the inverse solution is not unique, but it depends on the value of f. For a fixed f, the solution is unique. Thus, it is not reasonable to compare the matrices too carefully. The purpose of the comparison presented here is to show that the DBSFM and DDBSFM solutions are quite similar with respect to the corresponding optimal control problem.

Note that even though the human operator model does not include the explicit measurement of the position signal, the control laws generated by the optimal controllers solved based on the weighting matrices shown in Figure 7.7 use the estimate of the third state, which corresponds to the rope angle. Thus, the resulting control laws seem intuitively reasonable although it is evident that the human operator can "measure" also the angle.

7.3.3 Results for nonlinear HMI

The parameters of the models described in Chapter 6 were estimated by using the data shown on the right in Figure 7.2. The time domain estimation was performed using the ensemble average of the input and output signals. The frequency domain estimation was performed based on the ensemble average of the individual frequency responses of the trials.

The parameter estimation of the models were performed similarly as in the previous linear HMI data case. The MOCM and the HWMOCM parameters were estimated by setting initial values equally as in the linear case. However, the dead time delay was changed to $\tau = 0.13$ s for DDBSFM and HWDDBSFM, and to $\tau = 0.2$ s for DBSFM. In addition, the initial value for the neuromotor time constant for DDBSFM was set to $a_n = 0.1948$ (corresponds to $\tau_n = 0.3$ s), and for HWDDBSFM to $a_n = 0.2775$ (corresponds to $\tau_n = 0.2$ s).

For the Hammerstein-Wiener type models the output saturation and the rate saturation were given explicitly. In the data set with nonlinear HMI, the saturation level $\delta_{\max} = 4$ and the rate saturation level $d\delta_{\max} = 8$.

In order to provide a baseline for comparison of the performance of the developed models, a simple ARX model was also identified for the nonlinear HMI data using the exact same method described in the linear case in Section 7.3.1. The best structure (out of eight order models) is an ARX model where the number of free parameters in the numerator was 8, and the input delay lag was 3.

Numerical time domain comparison of the identified models against the validation data is shown in Table 7.3. In addition, also the number of free parameters for each model is shown. Similarly as in the linear case, the values for the performance indices are obtained by using

Table 7.3: Performance comparison and the numbers of free parameters of models for nonlinear HMI data.

Model	$V(\theta)$	$V_{\text{pos}}(\theta)$	$V_{\text{angle}}(\theta)$	N_θ
MOCM	18.3629	0.7591	0.0497	16
DBSFM	4.6026	0.3379	0.0192	15
DDBSFM	3.8305	0.1591	0.0177	14
ARX883	8.1116	0.1400	0.0269	17
HWMOCM	11.8804	0.4610	0.0302	16
HWDDBSFM	3.6461	0.2208	0.0255	14

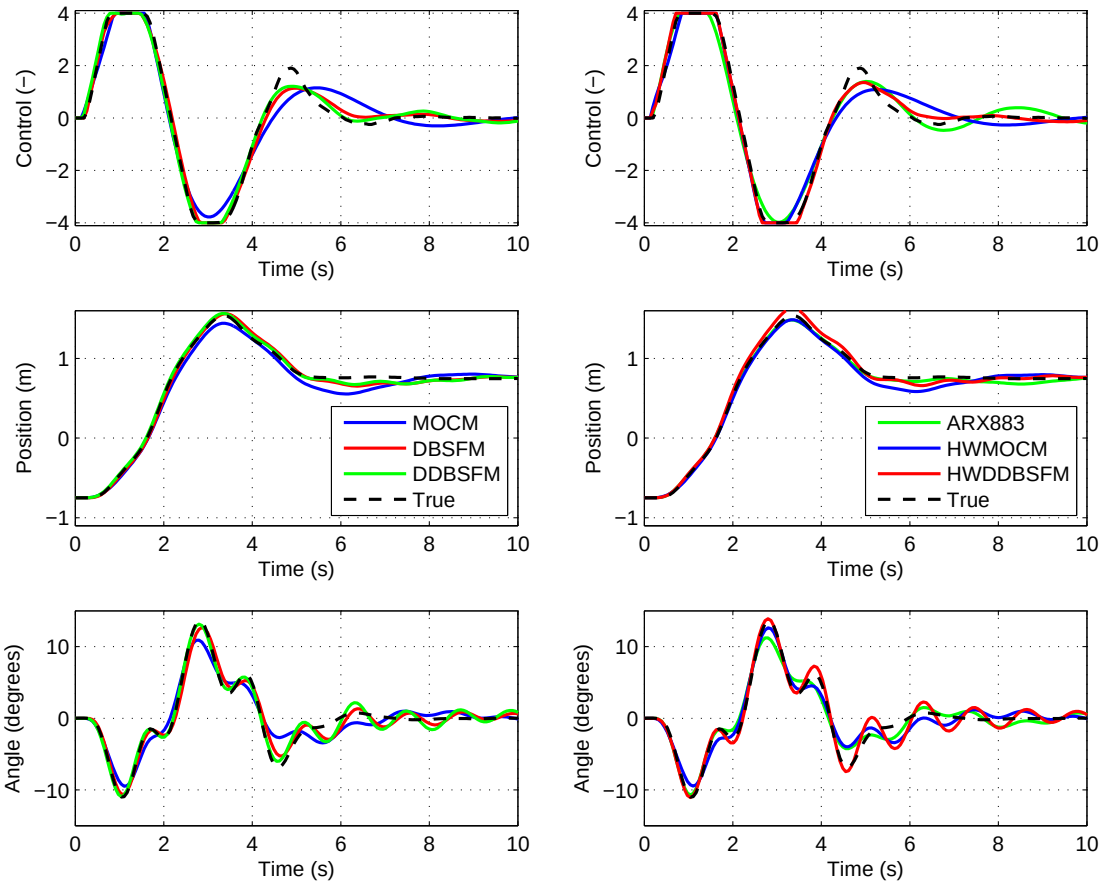


Figure 7.8: The time domain responses for nonlinear HMI data, given by the identified models when the model is set to execute the operator's task by the nonlinear simulator. The black dashed line represents the ensemble average of the measured signal based on the validation data.

the simulated responses shown in Figure 7.8. It must be emphasized that the simulator uses the same nonlinear HMI as in the case when the human operator controls the system. The signals shown in Figure 7.8 describe the control signal fed to the trolley crane. Thus, they have already been fed through the level and rate saturation nonlinearities.

According to the numerical results with respect to $V(\theta)$, the HWDDBSFM performs best, then DDBSFM, and then DBSFM. The Hammerstein-Wiener version of the MOCM also performs better than the conventional MOCM. The same conclusions can also be drawn from Figure 7.8. With respect to $V_{\text{pos}}(\theta)$ the best model is ARX883 and with respect to $V_{\text{angle}}(\theta)$ the best is DBSFM. The HWMOCM is significantly better with respect to all metrics than the regular MOCM. Based on the results one can conclude that the developed methods can simulate the human performance in accuracy comparable to the conventional linear models such as ARX. It seems that in terms of the prediction of the human operator control signal, the inclusion of the Hammerstein-Wiener structure in this case increased slightly the accuracy of DDBSFM but significantly the accuracy of MOCM.

7.4 Discussion

The structural human operator modeling methods described in Chapter 6 were implemented and tested in two experiments. In the experiments, the objective was to find methods for modeling the human operator in a case when the measurements are incomplete, meaning that all variables the human operator uses to control the system were not measurable. In addition, the human operator model should be able to retain the mapping from the parameters of the controlled machine to the parameters of the human model. This property is desirable for the human operator modeling in crane applications since the operating points of the crane vary.

In the experiment, the task of the human was to control the position of the crane and to regulate the angle of the load. However, it was assumed that the angle measurement is not available. Only the position of the crane and the human operator's control signal (after the human machine interface) were used. As it was discussed previously, this assumption is, in general not valid when considering the information the human uses to control the machine. Although the human operator models identified in this chapter do not explicitly use the angle information, they use the estimate of the angle in the control law, as it was seen in the control weighting matrices of MOCM, DBSFM and DDBSFM.

In the first experiment, the gain of the HMI is set so high that no significant saturation occurred. Nevertheless, in practice it turned out that in most individual trials, saturation occurred. In addition to the structural models, the human operator modeling is performed by using the best polynomial models for the same data set (as described in the author's publication [157]), that is, Autoregressive with Exogenous Input (ARX) and Hammerstein-Wiener Output Error Model (HWOE). The Hammerstein-Wiener models performed best according to the time domain performance criterion. The performances of the structural models identified using the gradient-based methods are comparable for the performance of the HWOE model and even better than the performance of the ARX model. However, an additional advantage for using the structural models is that they retain the mapping from the parameters of the controlled system to the human operator model. Thus, if the parameters of the system such as the mass of the load in a crane change, the human operator model changes also.

In the second experiment, the saturation level is decreased. In addition, the rate saturation is added to the HMI. Again, the Hammerstein-Wiener structure provided better performance in time domain for the MOCM and for the DDBSFM. The HWDDBSFM performed only slightly better than the DDBSFM, but the HWMOCM performed significantly better than the MOCM. The same observation can be made from the results obtained in the first experiment. The Hammerstein-Wiener structural models performed better in prediction of the human operator control signal than a linear ARX model in the nonlinear HMI case.

According to the results of both experiments, the use of the Hammerstein-Wiener structure is useful to obtain better simulation performance when the HMI contains nonlinearities. Especially for the polynomial models, the HWOE provides significantly better results than ARX. Here it should be pointed out that also the linear output error model structure was tried during the work done for the publication [157], but the results were not good. In the experiment for the nonlinear HMI, the input data used in the identification for all models were recorder after the nonlinearity. In other words, the human control signal had already passed through the level and rate saturation. Thus, also the linear models could exploit that information. If the human control signal before the nonlinearity was used, the Hammerstein-Wiener models would probably performed significantly better. However, it can be assumed that in real-life the signals after the nonlinearity are available and thus the experimental case presented in this thesis is justified.

For the DBSFM the inverse optimality criteria can be explicitly expressed in the frequency domain by using the return difference condition. Thus, the existence of the inverse optimal solution for the control and estimation problems can be guaranteed. In the experiment, it was confirmed that the return difference conditions for the DBSFM were fulfilled. Thus, the inverse solution with zero crossterms could be found by using the LMI-based approach. The inverse optimal solutions return the DBSFM to the optimal control framework and thus the adaptivity for the change in the dynamics of the controlled machine can be realized using the Riccati equations similarly as it is done in the conventional MOCM.

As it was discussed in Section 6.7, the corresponding inverse optimality criterion for the discrete time does not exist. Thus, the existence of the inverse optimal solution with zero crossterms (and positive semi-definite $\mathbf{Q}_0$ and $\mathbf{W}_1$) cannot be guaranteed for the DDBSFM. However, for nonzero crossterms the solution can be found if the state feedback and the estimator gains produce a stable controller and estimator. Thus, in the experiment, the inverse optimal control problem is solved for the DDBSFM with nonzero crossterms. The resulting optimal control solution provides the controller and the estimator gains equal to the gains obtained in the parameter estimation phase.

The DBSFM, DDBSFM and HWDDBSFM trained using the gradient-based approach outperform the MOCM and HWMOCM because the free parameters are directly the control and the estimator gains, not the weighting and the intensity matrices. Equal results would be achieved using the MOCM if all upper triangular elements of the weighting and the intensity matrices were set as unknown. However, this increases the number of unknowns so that the parameter estimation procedure becomes very slow.

In general, based on the experiments one can conclude that the structural models are able to model the human behavior in the point-to-point task equally or better than the simple polynomial models or polynomial-type Hammerstein-Wiener models. In conclusion, let us

consider again the modeling objectives shown in the beginning of this chapter.

The first objective (adaptability to the operating points) is fulfilled for MOCM and DBSFM, which can be adapted to the operating point changes using the Riccati equations. Since only the suboptimal inverse solution for DDBSFM could be found, it is not intuitively clear whether the DDBSM provides reasonable controller models when the operating points are varied. In comparison to the conventional simple models, the structural models studied in this thesis model the human operator with sufficient accuracy (the second objective), and the parameters could be successfully estimated using the experimental data (the third objective). Since the studied structural models are of high order, their parameter estimation methods are not as efficient as the ARX identification, for example. Especially for higher order controlled machines, the current identification solutions for the proposed models are not very fast. The fourth objective can be fulfilled using the Hammerstein-Wiener structure.

In the future, at least the following points should be addressed in the modeling methods:

- How well does the MOCM/DDBSFM models predict the human performance in different operating points when the model is identified using data from only single operating point?
- Could the inverse optimality criteria for the DDBSFM be stated so that they could be used as constraints in the parameter identification process?
- Can the suboptimal inverse solution of DDBSFM adapt to the operating point changes, even in a small range of machine parameter values?
- Could the feedforward and intermittency studied in [41, 70] and the switching control structures studied in [115, 114] be included in the human operator model. How could the HMM-based work cycle modeling be exploited here?
- Could the structured approach for data-based learning of inverse optimal control laws described in [162] increase the identification efficiency of the structural models studied in this thesis?

Before having truly useful and efficient operating point adaptive human operator models with the feedforward structure, the above points should be answered first. Within this thesis, the points could not be studied further.

It is worth to mention that the adaptability to operating conditions can also be obtained using classical crossover models [99, 98]. However, these models are accurate only in a small range of frequencies near the crossover frequency [34]. In the future, a comparison of the performance of the crossover models and the optimal control type models in crane control should be done, especially from the adaptivity point of view.

8 Human skill adaptive HMI

The fifth interaction level of HAMC described in Chapter 3 is *passive adaptation*. In this thesis, passive adaptation refers to a property where the machine adapts itself to the human operator's skill level or control characteristics but does not actively intervene to the human operators control signal (in contrast to the active adaptation). Passive adaptation, in general, can be realized using several approaches. Analytic approach is to use the models of the human operator, the HMI and the controlled machine to determine the optimal HMI parameters according to some design criteria. On the other hand, knowledge-based approach is to use the skill evaluation results described in Chapter 5 together with the knowledge of the current parameter setup of the machine. Using that information, a dedicated reasoning system evaluates whether the current parameter setup is suitable or not. The reasoning system could have the same structure as the ICS for working machines described in [119]. The passive adaptation could as well be implemented utilizing the assist control tuning method proposed in [146, 145], where the tuning of the virtual internal model (corresponds to the HMI here) is based on the Lyapunov functions.

The virtual internal model modification allows to achieve a desired behavior of the controlled system based on the external measurements [84]. It is generally used in the existing HAM systems and has proven to provide excellent results. In this thesis, the tuning of the HMI is posed as a simple controller tuning problem using the models of the human operator, the HMI and the controlled machine. If only the HMI sensitivity tuning is considered, the method proposed in this thesis is very easy to be implemented.

In general, the human performance is time-varying and stochastic. In addition, the human operator adapts to the changes in the HMI and he/she also learns while repeating similar tasks several times. Therefore, the tuning of the HMI should be done little-by-little. That is, once the human model for the current settings has been identified, or alternatively the skill/performance for the current settings has been evaluated, the optimal parameters for the HMI can be found. Once the parameters are changed, the human model/skill evaluation should be performed again using the data gathered from the changed system. Then using the new model, again the new optimal parameters for the HMI can be found. This procedure is repeated while the parameters of the HMI have settled to a stable level. In this thesis, the above procedure is called Skill Adaptive Control (SAC) algorithm and it is shown in the experiment in Section 8.6 that for the test subjects the algorithm converged towards a value where the task performance is rather constant. The SAC algorithm is designed to be applicable for adapting crane-like systems to the human skill level based on normal operation data.

While the human operator modeling problem is studied from the perspective of structural models in Chapters 6 and 7, in this chapter a general tuning problem is considered. That is, the tuning methods should be valid for the human model identified using any method that provides continuous approximation of the human behavior as a controller. In the experiments for the recursive SAC tuning, where the operating point is kept constant, a linear DDBSFM is used.

This chapter presents a general structure of a system which can be used to realize the passive adaptation interaction level of the proposed HAMC. In particular, it is described how the HMI adaptation can be interpreted as a classical controller tuning problem, when the human

operator and the machine models are known. The content of this chapter is mainly based on the author's publication [154]. However, the HMI adaptation as a controller tuning problem is explained more thoroughly here. As an alternative method for the analytic approach for the HMI adaptation, the knowledge-based approach is described in the case of HMI sensitivity tuning.

8.1 Description of the HMI adaptation system

In the area of flight control design, there exists widely accepted procedures to design the control interfaces so that the stability of the aircraft is retained even in the worst case. A good summary of the flight control design is given in [11]. To prevent the loss of stability and the pilot induced oscillations well-known describing function based nonlinear analysis and the robust control theory can be used. However, whereas the flight control for obvious reasons aims at optimizing the stability for *any human operator*, the objective of the human adaptive system is to optimize the maneuverability and the performance of the machine for *an individual human operator*. The HMI adaptation realizes the passive adaptation interaction level of the HAMC described in Chapter 3.

The idea of the HMI adaptation is to change the control interface to better suit for the operator's current skill level. The objective in the tuning is to prevent the unwanted phenomena such as oscillation due to the insufficient skill level of a human operator. Simultaneously the aim is to maximize the overall long term performance of the system. In an industrial process, such as a gantry crane, the HMI adaptation includes the phases shown in Figure 8.1. The structure shown in the figure is based on the framework for intelligent coaching of machine operators [119] (Figure 3.6).

The first step is *intent recognition*, which can be performed using the HMM-based method descsribed in Chapter 4. Once the intent of the operator has been recognized, the objective of the work is known. The next step is to evaluate or model the task execution. This step is called *skill evaluation/human model identification*. In this thesis, skill evaluation is studied in Chapter 5 and human model identification in Chapters 6 and 7. In this phase, the operator's skills are evaluated or alternatively the skill is modeled as a mathematical model of the human controller.

Given the current skill level (or the human model), the current parameter setup of the man-machine interface, and the model of the system, one can find the optimal parameter configuration for the skill level. In practive, the HMI parameter optimization is implemented as a subroutine of the intelligent coaching, assistance and adaptation block of Figure 3.1. In the analytic approach, the *parameter optimization* step consists of the solution of the optimization problem

$$\boldsymbol{\xi}^* = \min_{\boldsymbol{\xi}} J(g_m(\mathbf{x}), g_p(\mathbf{x}), g_i(\boldsymbol{\xi})), \tag{8.1}$$

where g_m represents the controlled machine, g_p represents a human operator, and g_i is a mathematical representation of the HMI as a function of parameters $\boldsymbol{\xi}$. The cost criterion J can be any well-defined design objective. Solution of the optimization problem produces the optimal parameters $\boldsymbol{\xi}^*$ for the HMI.

In the knowledge-based approach, the parameter optimization is an expert system which per-

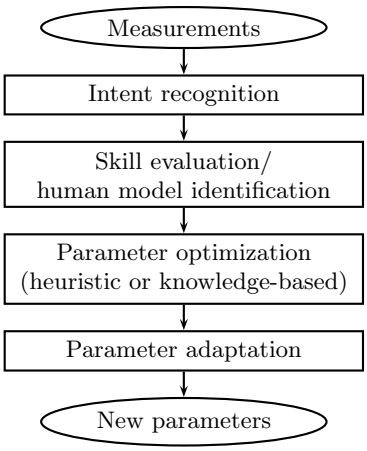


Figure 8.1: A flowchart of the phases in a general HMI adaptation system.

forms reasoning about the suitability of the current parameter setup $\boldsymbol{\xi}$ based on the skill evaluation results $\mathbf{Z}$

$$\boldsymbol{\xi}^* = \text{HeuristicOptimization}(\mathbf{Z}, \boldsymbol{\xi}), \tag{8.2}$$

where "HeuristicOptimization" is, for example, an FIS. This approach is feasible especially when no accurate human model is available.

In *parameter adaptation* phase the parameters $\boldsymbol{\xi}^*$ are used to define a new parameter setup. The method of adaptation depends on the application. One can use, for example an iterative learning rule to update the parameter setup of the machine. However, the adaptation should be slow enough so that the usability of the system would not suffer. The concept of JND can be used to determine the maximum allowable change in the parameters [63].

The human skill adaptive HMI system can be seen as a block diagram shown in Figure 8.2. The human controller tries to control the process based on the desired states (the intent) and the measured states (the observations given by the human senses). The human operator has a skill level which can be evaluated based on the recognized intent, measured states, and the control actions introduced by the operator. Alternatively, the skill is abtracted into a mathematical model describing the human control characteristics. The operator's control actions are fed through a human-machine interface (HMI) which scales/filters the control signals given to the actual process to better suit the motor-sensory skills (or the control characteristics) of the operator. The HMI block includes the solution of the optimization problem (8.1) or (8.2).

8.2 HMI adaptation as a controller tuning problem

8.2.1 General description

Assuming that the human operator can be approximated using linear models with sufficient accuracy, and that the model of the controlled machine is known, the human adaptivity can be

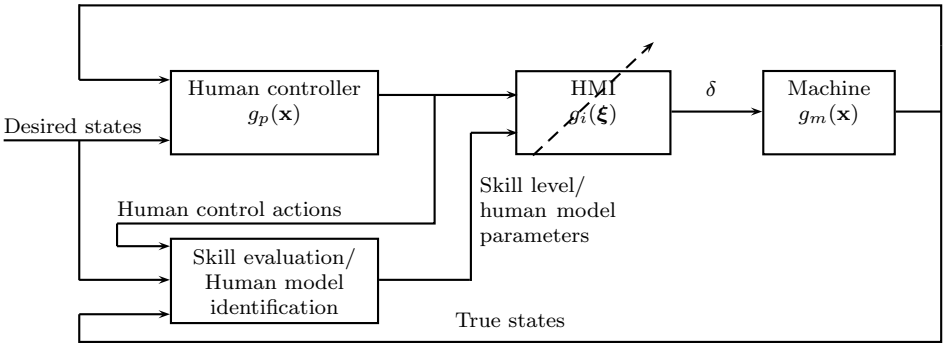


Figure 8.2: A block diagram of the HMI adaptation system.

seen as a classical controller tuning problem. We are given a machine $g_m(\mathbf{x})$, and a controller $g_p(\mathbf{x})g_i(\boldsymbol{\xi})$, where $g_p(\mathbf{x})$ is the human operator (unmodifiable part) and $g_i(\boldsymbol{\xi})$ the HMI (modifiable part). The tuning task is now to find such values for the free parameters $\boldsymbol{\xi}$, which fulfill the design criteria for the controller tuning.

In other words, considering the linear systems in the Laplace domain, the problem is to determine the parameters $\boldsymbol{\xi}$ of the HMI G_i in the closed-loop system

$$G_{cl}(s) = \frac{G_p(s)G_i(s,\boldsymbol{\xi})G_m(s)}{1 + G_p(s)G_i(s,\boldsymbol{\xi})G_m(s)} \tag{8.3}$$

to fulfill the design criteria. To solve the tuning problem, any method known from the classical control theory could be used, but in the following, three methods are described.

8.2.2 Special case: sensitivity of the HMI

When a human operator controls a machine with e.g. joysticks he/she tends to use the maximum available control range. Therefore, if the gain of the HMI is too high (interface is too sensitive), the operation becomes very difficult as the system reacts rapidly to even a small accidental movement of the joystick. Thus, the gain of the control interface should be tuned so that the response of the system to the joystick movements would correspond to the human operator's ability to predict the system behavior, to estimate the states of the system, and the time delay and time constant of the human neuromuscular system. On the other hand, if the gain is too small for the operator's skill level, the performance is not optimal, because the operator could handle faster movements and thus achieve better performance.

In terms of control engineering, the gain optimization problem is very similar to the familiar P-controller design problem. If the current gain of the man-machine interface is denoted by k_0, the new gain can be written as proportional to the current gain. That is, the man-machine

interface becomes simply

$$G_i(s, \boldsymbol{\xi}) = \frac{k_n}{k_0}, \tag{8.4}$$

where k_n is the unknown constant parameter. Now the task is to find k_n which fulfills the design criteria for the closed-loop performance.

8.2.3 Tuning methods for HMI adaptation

8.2.3.1 Numerical optimization based tuning

The optimization based tuning is a conventional method for tuning controllers and it is applicable especially for nonlinear problems. In the optimization based tuning, the controller design problem is converted into an optimization problem, where the task is to minimize a cost criterion (or several cost criteria) using the free tuning parameters. In the case of HMI adaptation, the tuning parameters are $\boldsymbol{\xi}$.

In the controller tuning, often ISE type of cost criteria are used. Using a nonnegative definite weighting matrix $\mathbf{Q}$, the ISE criterion can be formulated as follows

$$J = \int_{t=0}^{T} \mathbf{x}_e^T(t)\mathbf{Q}\mathbf{x}_e(t)dt, \tag{8.5}$$

where $\mathbf{x}_e(t)$ ($\mathbf{x}_e(t) \in \mathbb{R}^{N_s}$) is the difference between desired state trajectory and the true state trajectory. By varying the weighting $\mathbf{Q}$, the minimization of (8.5) results in a closed-loop response with different properties. However, because the model of the human operator is not perfect (8.5) might favor parameters which lead to a response which is too fast for the human operator.

By introducing a term $\dot{\delta}^2(t)$, which punishes the rapid control movements, one can find a set of parameters which provide a smooth and fast response. Thus, the modified criterion can be written as

$$J = \int_{t=0}^{T} \left(\mathbf{x}_e^T(t)\mathbf{Q}\mathbf{x}_e(t) + f\dot{\delta}^2(t) \right) dt, \tag{8.6}$$

where f is the weighting coefficient for the derivative of the control signal. In particular, because the criterion is in the same form as the optimal control models (OCM, MOCM, and the inverse solutions of DBSFM and DDBSFM) assume, the operator's individual weights obtained in the human operator identification can be used as the weighting for (8.6).

8.2.3.2 Optimal tuning approach

If the human operator model is of the optimal control type (e.g. MOCM, DBSFM with inverse solution etc.), the optimal control theory based tuning approach can be used. The optimal theoretical value of the control performance index can be derived based on the stochastic control theory. Therefore, one can study if the performance index can be decreased by fixing the operator model (as it is) and then changing the control interface. Here, only the HMI sensitivity optimization is presented, but the optimal tuning approach could be used to tune more compex filters of HMIs as well.

As described in Appendix B.2, for optimal control type of models the numerical value of the performance index (of the 1-plant) is given by

$$J_p = \mathrm{tr}\left(\tilde{\mathbf{X}}\tilde{\mathbf{R}}\right), \tag{8.7}$$

where $\tilde{\mathbf{X}}$ is the solution of the Lyapunov equation

$$\tilde{\mathbf{A}}\tilde{\mathbf{X}} + \tilde{\mathbf{X}}\tilde{\mathbf{A}}^T + \tilde{\mathbf{V}} = 0, \tag{8.8}$$

and $\tilde{\mathbf{A}}$, $\tilde{\mathbf{V}}$, and $\tilde{\mathbf{R}}$ are given Appendix B.2 for MOCM. Only now the plant model is replaced with the changed system model to obtain

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{1,n} & -\mathbf{B}_1\mathbf{l}_1 \\ \mathbf{F}\mathbf{C}_1 & \mathbf{A}_1 - \mathbf{B}_1\mathbf{l}_1 - \mathbf{F}\mathbf{C}_1 \end{bmatrix} \qquad \mathbf{A}_{1,n} = \begin{bmatrix} \mathbf{A} & \frac{k_n}{k_0}\mathbf{B}\mathbf{C}_d & \frac{k_n}{k_0}\mathbf{B} \\ 0 & \mathbf{A}_d & \mathbf{B}_d \\ 0 & 0 & -\frac{1}{\tau_n} \end{bmatrix} \tag{8.9}$$

when the HMI adaptation property is the sensitivity (gain) of the HMI as in (8.4). Now the HMI gain optimization problem can be stated as

$$\begin{aligned} &\min_{k_n} \mathrm{tr}\left(\tilde{\mathbf{X}}\tilde{\mathbf{R}}\right) \\ &\text{s.t.} \begin{cases} \tilde{\mathbf{A}}\tilde{\mathbf{X}} + \tilde{\mathbf{X}}\tilde{\mathbf{A}}^T + \tilde{\mathbf{V}} = 0 \\ \text{and } \tilde{\mathbf{A}} \text{ as in (8.9)} \end{cases}, \end{aligned} \tag{8.10}$$

which can easily be solved using numerical optimization procedures. Now the control rate weighting f and $\mathbf{Q}_y$ act as a design parameters. Larger values of f result in a smaller gain and smoother response. The weighting matrix $\tilde{\mathbf{R}}$ (as a function of f and $\mathbf{Q}_y$) is given in Appendix B.2.

8.2.3.3 Frequency domain tuning

The frequency domain tuning is a classical method to tune linear controllers. For a linear SISO system, the phase margin is related to the closed-loop specifications of the control performance, such as the overshoot percentage or the damping ratio (of the dominant poles). A control system with low phase margin will have a larger overshoot and more oscillation than with a high phase margin.

The phase margin can be used as a tuning criterion for the HMI adaptation as follows. The phase margin of the closed-loop system including the human, the man-machine interface, and the machine (provided that the open-loop system is stable) is

$$PM = \arg\left[G_p(j\omega_0)G_i(j\omega_0;\boldsymbol{\xi})G_m(j\omega_0)\right] + 180^\circ, \tag{8.11}$$

where G_p, G_i, and G_m are the frequency functions of the human operator, the man-machine interface, and the controlled machine, and ω_0 is the frequency at which the gain of the open loop transfer function is zero dBs. The controller tuning task is now to find such parameters $\boldsymbol{\xi}$ which produce desired phase margin. In order to have a decent closed-loop performance

(overshoot less than 10 %), at least 60° phase margin is needed (based on the second order approximation of the dominant poles) [93].

The phase margin criterion presented here is similar to the well-known crossover model principle [99]. In the crossover model, the open-loop human-machine system is assumed to behave according to

$$G_p(j\omega)G_i(j\omega)G_m(j\omega) \approx \frac{\omega_0}{j\omega}e^{-j\omega\tau_p} \tag{8.12}$$

around the crossover frequency ω_0. The phase margin (in radians) of the human-machine system is now approximately

$$PM = \frac{\pi}{2} - \omega_0\tau_p. \tag{8.13}$$

When the human operator model is known, the HMI $G_i(s)$ can be tuned so that that the bandwidth ω_0 is maximized and the phase margin is kept higher than the desired phase margin.

To account for the uncertainty in the human operator and plant modeling, as well as the changing environmental conditions (e.g. dynamics of the manipulated objects), the robust control theory could be used in the HMI adaptation. In [128], the robust control method was applied for tuning of a telemanipulator. The same method could also be applied here. However, for simplicity, only phase margin criterion is used in this thesis.

In case of nonlinearities, such as saturation or rate limiter, the frequency domain analysis becomes difficult. One can use describing functions to analyze the behavior of the nonlinear system with predefined magnitudes of the input signals, but in general, the nonlinear frequency domain analysis is tedious.

8.3 Knowledge-based approach for HMI adaptation

As an alternative method of adapting the HMI with respect to the human skill and dynamic characteristics, one can think of using knowledge-based or heuristic methods, such as Fuzzy Inference Systems (FISs). This approach can be considered especially when no accurate human model is available. In this case, the adaptation is based on similar reasoning shown in the ICS structure in [119]. In this thesis, FISs are used to implement the reasoning how the HMI parameters should be tuned in the knowledge-based approach, because they provide convenient means to imitate the reasoning of a human expert. The theoretical foundation of the fuzzy logic and FISs is strong [175, 74] and there are widely used tools available to implement FISs.

When tuning purely the sensitivity of the HMI, the reasoning system becomes very simple. Consider a Sugeno-type FIS, where the inputs are the probability of high performance (PrHighPerformance), the current gain of the HMI (CurrentGain), and the information whether the operator uses the full control range available to perform the task (FullControlUse). The FullControlUse can be defined as

$$\text{FullControlUse} = \frac{\max_t |\delta(t)|}{\delta_{\max}}, \tag{8.14}$$

where $\delta_{\max}$ is the maximum available control signal.

The first input describes the probability to obtain high performance with respect to any perfor-

mance or skill index. The index value can be interpret as probability, because the skill indices are scaled using the statistical learner model described in Section 5.4. The output of the FIS is GainSuggestion, which describes whether the gain of the HMI sould be increased, decreased or kept constant. The output has then three constant membership functions: DECREASE (value -1), OK (value 0), and INCREASE (value 1). Assume two-level partition for each input variable (LOW and HIGH), where the membership functions can be of any type. Now, the knowledge-based tuning can be realized simply by using the following rulebase.

1. If (PrHighPerformance is not HIGH) and (CurrentGain is not LOW) then (GainSuggestion is DECREASE)
2. If (PrHighPerformance is not HIGH) and (CurrentGain is not HIGH) and (FullControlUse is HIGH) then (GainSuggestion is INCREASE)
3. If (PrHighPerformance is HIGH) then (GainSuggestion is OK)

The aim of the first rule is to decrease the gain if the low performance is due to a too high gain value. The second rule increases the gain if the low performance is due to a too low gain value. In addition, it is required that the operator uses full control range until the gain is increased. This requirement is made because if full controls are not used, the operator could increase the performance by using larger control lever motions. If the performance is already at a high level, no modifications for the current HMI are needed.

The defuzzification of the FIS is performed using the weighted sum method, because it does not normalize the sum of the outputs to unity. If a new data point does not fit to the rulebase well, all outputs obtain low weights. The better the data point fits to the FIS rulebase, the higher the confidence of the decision and thus higher changes in the gain can be allowed.

By using the knowledge-based approach, the suggestion how the sensitivity of the HMI should be modified is given by

$$\Delta k_n = \alpha k_n \mathrm{FIS}\left(\mathbf{Z}, k_n\right), \tag{8.15}$$

where $\mathbf{Z}$ the vector containing the inputs for the FIS and $0 \leq \alpha_i \leq 1$. The parameters more suitable for the current performance level of the operator are obtained using

$$k_n^* = k_n + \Delta k_n. \tag{8.16}$$

In (8.15), the FIS gives values in between -1 and 1, which can be interpreted as the relative change of the gain. When multiplied with the current value k_n, the absolute change is obtained. The scalar α can be used to control the parameter adjustment.

8.4 Skill adaptive control (SAC) algorithm

The tuning methods proposed above find the parameters of the HMI which are suitable for the current model of the human operator with respect to the tuning criteria. However, as it has become clear earlier in this thesis, the human operator adapts to the change in the machine's dynamics as well as to the change in the HMI. Thus, the solution given by the tuning methods cannot be considered global. That is, if the parameters of the HMI are changed, the human

adapts himself/herself to the changed system. As a result, the human operator's dynamic model changes and the HMI parameters may not anymore fulfill the design criteria. Therefore, in the following, a recursive approach called Skill Adaptive Control (SAC) algorithm is desribed. The idea is that the solution for the HMI tuning problem given by the tuning methods is used only as a direction where the parameters should be changed. The magnitude of the parameter change depends on the amount the improvement to be achived due to the change.

For simplicity, the following assumptions are made for realizing the SAC algorithm. It is assumed that the operator performs K trials of a well-defined control task. Moreover, the task the operator executes is same among the trials, and the system dynamics remain constant. Thus, the plan of the operator is known beforehand. After performing K trials, the operator's dynamics are identified (analytic) or the skills are evaluated (knowledge-based). Thereafter, the optimal parameters for the HMI are searched. Finally the parameters of the true HMI are adapted to better suit the operator's current skill level. By using the new parameters, K trials are performed again and so forth.

In the analytic method, the HMI controller tuning problem is solved by using the mathematical model of the controlled machine $G_m(s)$ and the identified mathematical model of the human operator $G_p(s)$. The human operator model can be identified using any method. Thereafter, any of the three design approaches can be used to obtain the optimal parameters $\boldsymbol{\xi}^*$ for the HMI. In the knowledge-based approach, the human operator's skills are evaluated using the methods of Chapter 5. The optimal parameters are obtained by using (8.2).

Once the optimal parameters for the current dynamics/skills of the human operator have been found, the parameters need to be adjusted so that the future performance is maximized. The parameters are adjusted by using the following iterative adaptation rule

$$\boldsymbol{\xi}(k+1) = \max\left(\min\left(\boldsymbol{\xi}(k) + \gamma(\boldsymbol{\xi}^*(k) - \boldsymbol{\xi}(k)), \boldsymbol{\xi}_{\max}\right), \boldsymbol{\xi}_{\min}\right), \tag{8.17}$$

where γ $(0 \leq \gamma \leq 1)$ is the learning rate, k is the iteration index, and $(\boldsymbol{\xi}_{\min}, \boldsymbol{\xi}_{\max})$ are the minimum and the maximum allowed values for the parameters. One needs to be careful when choosing the learning rate, since (in the analytic case) the operator's model is identified based on K number of trials. If K is too small, the model does not necessarily represent the operator's dynamics well, and as a result, the solution of the optimization problem (8.1) might be incorrect. In the knowledge-based approach, too small K enables high variation of the average performance and thus variance in the decision of the reasoning system.

The proposed implementation of the SAC can be put in an algorithmic form as follows.

1. *Initialization*: Set $k = 1$ and the initial values $\boldsymbol{\xi}(1)$.

2. *Data gathering*: Perform K trials of a given task.

3. *Identification/evaluation*: Identify the mathematical model of the human operator based on the data gathered during the trials (analytic approach). Evaluate the performance with respect to the chosen criterion and update the associated GEV distributions[17] (knowledge-based case).

[17]Updating the distributions is not necessary but if this is done, one obtains self-tuning parameters for the evaluation.

4. *Optimization*: Solve (8.1) or (8.2) to obtain the optimal system parameters $\boldsymbol{\xi}^*(k)$ for the current operator dynamics.

5. *Adaptation*: Adjust the current system parameters to better suit the operator's skill level by using (8.17).

6. *Termination*: Set $k = k + 1$ and go back to step 2. Alternatively, the algorithm can be terminated if a predefined stopping criterion is fulfilled.

In order to prevent the algorithm to change the parameters of the HMI, when the new parameters do not produce better control results, the following rule can be used for the learning rate in the analytic approach.

$$\gamma = \min\left(\max\left(1 - \frac{J(\boldsymbol{\xi}^*(k))}{J(\boldsymbol{\xi}(k))}, \gamma_{\min}\right), \gamma_{\max}\right) \tag{8.18}$$

Thus, the parameters are adjusted only when the new parameters actually lead to a better value of the control design criterion J, assuming that the criterion value is decreasing with the goodness of the design. In the knowledge-based approach, the step size is automatically controlled if the FIS is built as described in Section 8.3. If the performance is already high, the FIS will not suggest large changes to the parameters.

8.5 Experiment: comparison of the tuning methods using simulation

For illustration, let us study the position transfer function of (7.2), that is,

$$G_m(s) = \frac{s^2 + 9.81}{s^4 + s^3 + 29.43s^2 + 9.81s} \tag{8.19}$$

and the HMI

$$G_i(s) = \frac{k_n}{8}, \tag{8.20}$$

where $k_n = 8$ in the linear case of the experiment described in Section 7.1. The numerical values for the MOCM identified for the linear HMI data are shown in (D.1) – (D.7). The corresponding transfer function of the MOCM becomes

$$G_p(s) = \frac{-414.50(8.89)(0.35)(12.19, 49.50)(-0.19, 40.06)}{(8.89)(10.70)(0.74, 8.39)(2.16, 46.13)(12.74, 88.03)} e^{-0.49s}, \tag{8.21}$$

where (x) describes the term $(s+x)$ and (x, y) the term (s^2+xs+y). Note that the gain of the model is negative. This is because sign of the data was "wrong". Therefore, when using (8.21), the sign of the gain should be changed.

By using the MOCM, the system model and the HMI model, the three tuning criteria were evaluated with respect to the sensitivity of the HMI. For the simulation tuning, and the cost function (8.6) parameters were set as $f = 0.001$ and $\mathbf{Q} = \mathbf{Q}_0([1, 2, \ldots, N_s], [1, 2, \ldots, N_s])$. The resulting cost function is shown in the leftmost plot of Figure 8.3. For the frequency domain tuning, there are no free parameters. Only the phase margin needs to be computed, which is shown in the middle plot of Figure 8.3. For the optimal Lyapunov-based tuning, the parameters

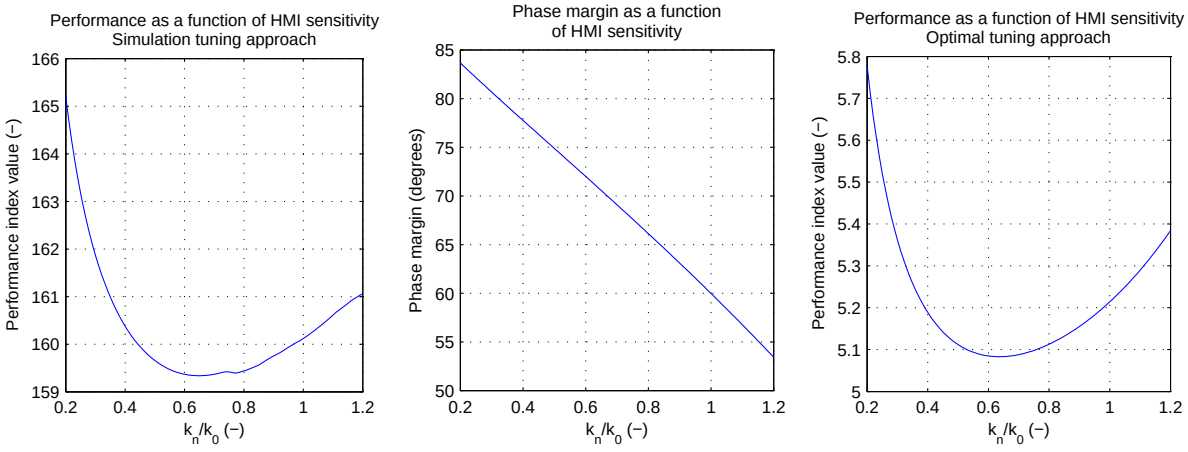


Figure 8.3: The values of the tuning criteria with the sensitivity of the HMI. Leftmost is the cost function for simulation tuning, middle the phase margin curve for frequency domain tuning, and rightmost the optimal Lyapunov tuning cost function.

were obtained directly from the MOCM, but f, which is needed in $\tilde{\mathbf{R}}$ was set to be $f = 0.004$. The resulting cost function is shown in the rightmost plot of Figure 8.3.

The results are shown in Figure 8.3. The optimal value for the ratio between the new gain and the current gain of the HMI is approximately 0.64 for the simulation tuning, 0.66 for the frequency domain tuning (assuming the phase margin 70°), and 0.64 for the Lyapunov-based optimal tuning. Here, of course, the parameters were chosen so that approximately equal results would be obtained. The objective was to show that all three methods can be used to tune the HMI.

The optimal tuning minimizes the quadratic cost criterion of a noise driven closed-loop human-machine system, where the noise characteristics are based on the identified human model. Since the noise covariances vary from human to human, the result of the optimization will be different for different human operators. Therefore, it is tested how the solutions given by the optimal tuning method vary with the noise levels. The same MOCM model and tuning parameters as previously is studied. However, the noise parameters $\mathbf{W}$, $\mathbf{V}_y$ and V_u are varied. In practice, the original matrices are multiplied by a scalar varying from 0.1 to 10. The solutions given by the optimal tuning method are plotted against the relative noise level in Figure 8.4 when one noise parameter at time is varied. In addition, the control result with the corresponding MOCM and the varied controllers are shown in Appendix E.

One can note that when the motor noise covariance V_u decreases, the suggested relative optimal gain increases. This is reasonable, because the motor noise covariance describes the "accuracy" of the human control signal. The corresponding simulation results are shown in Figure E.1, where the response simulated by the optimal relative gain value is shown red. That strong relationship is not found in this case, when the measurement noise covariance matrix $\mathbf{V}_y$ is varied. First the solution decreases slightly and after $k_n/k_0 = 0.3$ it starts increasing slowly.

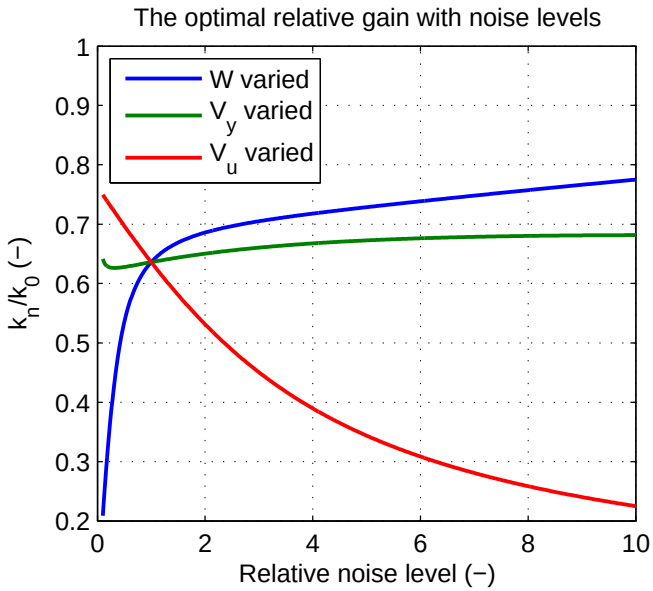


Figure 8.4: Evolution of the optimal relative gain given by the optimal tuning approach as a function of relative noise level.

The simulation results are shown in Figure E.2 When the plant noise covariance matrix $\mathbf{W}$ is varied, the increase in the plant noise leads to higher suggested relative gain values. This is also reasonable, because the increase in the plant noise increases the speed of the Kalman filter. The simulation results for the plant noise covariance variation are shown in Figure E.3.

The choices of the weighting parameters for the simulation-based tuning and the optimal tuning affect significantly to the results. The advantage of the frequency domain approach is that one can design the closed-loop system to achieve desired response statistics, such as overshoot. Moreover, the frequency domain and the optimal tuning methods are not suitable for nonlinear models or HMIs. Thus, the choice of the tuning criteria as well as the tuning parameters depend on the used human, HMI and machine models.

8.6 Experiment: iterative SAC tuning

8.6.1 Setting up the experiment

In order to test the SAC algorithm, the trolley crane simulator described in Section 7.1 is used. The task of the operator is the same as in the previous experiments, that is, to transfer the cart from one point to another and then compensate the swinging of the load.

The objective of the experiment is to tune the gain (i.e. the sensitivity) of the control interface using the SAC algorithm. In this experiment, the analytic tuning with the phase margin

criterion is used to obtain the suitable parameters as follows.

$$k_n^* = \max k_n, \text{ s.t. } PM \geq 63^\circ, \tag{8.22}$$

where k_n^* and k_n are defined as in (8.4). The phase margin limit of 63° is chosen because it allows some overshoot, which, based on the experiments is typical for a human operator performing a point-to-point type task with a crane. The solution of the problem (8.22) results in a gain k_n which is suitable for the current state of the dynamics of the human operator given the prevailing task and machine dynamics.

8.6.2 Implementation of the SAC

To identify the human controller based on the K trials of the given task, the DDBSFM method is used. The initial values for the parameter estimation are chosen to be equal with the DDBSFM identified in Section 7.3.1. The identification is carried out using the gradient-based sequential quadratic programming algorithm implemented in Matlab Optimization Toolbox routine "fmincon". Because the DDBSFM does not include the input delay lag as a free parameter, the identification procedure is performed for several (reasonable) candidates of the input lag. Finally the lag resulting in the best performance is chosen.

The SAC tuning algorithm was implemented so that the operator performed $K = 10$ trials of the point-to-point task, after which the human controller model was identified and the suitable HMI parameters were searched. For the next round of K trials, the new gain of the HMI was set according to

$$k_n(k+1) = \max\left(\min\left(k_n(k) + \gamma(k)\left(k_n^*(k) - k_n(k)\right), 9.0\right), 0.5\right), \tag{8.23}$$

where

$$\gamma(k) = \gamma_2 \min\left(\max\left(\frac{|PM\left(k_n^*(k)\right) - PM\left(k_n(k)\right)|}{5}, 0.01\right), 0.5\right). \tag{8.24}$$

The maximum γ value 0.5 is chosen to prevent too high sudden changes in the HMI sensitivity. On the other hand, it provides fast enough convergence for the SAC algorithm with the chosen number of trial rounds. The minimum γ value 0.01 is chosen so that at least small changes in the sensitivity are allowed even though the phase margin is close the optimal phase margin. The role of γ_2 can be explained as follows. Based on the experiments, the beginners tend not to use the full control range. This resulted in high phase margins and thus the algorithm increased the sensitivity although the full control range is not used. Thus, γ_2 is defined as

$$\gamma_2 = \begin{cases} 1, & \text{if} \quad \max_t \frac{|\delta(t)|}{k_n(k)} \geq 0.98 \\ 1, & \text{if} \quad \frac{k_n^*(k)}{k_n(k)} \leq 1 \\ 0, & \text{otherwise} \end{cases}, \tag{8.25}$$

that is, the operator should exploit at least 98 % of the full control range in order to allow the system to perform the adaptation towards the higher gains. If the suggested gain is lower than the current gain the gain should always be decreased.

8.6.3 HMI adaptation results

The SAC algorithm is run by four operators, who knew the purpose of the experiment. The first operator, OP1 performed the SAC iteration starting from a very low initial value for k_n ($k_n = 0.6$) as well as from a high initial value ($k_n = 8$). The other operators OP2, OP3 and OP4 performed the iteration starting only from low inititial value ($k_n = 0.6$). Operator OP1 is a very experienced on this task. Operator OP2 is a beginner, with the least amount of practice. Operators OP3 and OP4 are rather experienced but with less training than OP1.

Each operator performed a total of 30 rounds of 10 trials, except for OP1 who performed the the experiments twice starting from different initial values. Thus, operators OP2, OP3 and OP4 performed the point-to-point task in total 300 times and operator OP1 600 times.

The results of the SAC algorithm experiment are shown in Figure 8.5. In the upper left plot, the true sensitivity (value of k_n) of the HMI is shown. Remember that the value is updated only once in each ten executions of the task based on the model identified for the past ten trials. In the upper right plot, the suggestions given by the phase margin method are shown. For clarity, the gain trajectories of the high initial gain case of OP1 are put in separate plots (the middle row). The value of the phase margin for the identified model computed using (8.11) are shown in the lower left plot. In addition, the plot shows the 63° design criterion as a dashed thick line. The average task execution times (average of each ten trials) are shown on the lower right plot. The average value is computed by removing the best and the worst values from the set of ten and then taking the average from the remaining eight values.

In the low initial value case of OP1, the algorithm starts slowly increasing the value of k_n because, on the average, the phase margin is greater than 63°. After 200 rounds of the task, the phase margin becomes so close to 60° that the convergence slows down significantly. The average task execution time decreases slightly as the gain of the HMI increases. In the high initial value case of OP1 (the gain curve shown in the middle left plot), the phase margin is initially less than 55°, and the average task execution time is almost eigth seconds, even though OP1 can be considered as an expert. After 15 rounds of 10 trials the phase margin stays over 60° and the convergence slows down. However, the gain seems to continue decreasing slowly. The average task execution time remains practically constant after the tenth round of 10 trials.

Operator OP2 is a beginner. In the first round, his average task execution time is more than 12 seconds. Although the phase margin is generally higher than the required 63°, the SAC algorithm does not allow to increase the gain, because the operator does not exploit the full control range. Within 30 rounds of the task, the value of k_n rises from 0.6 to about 0.75. It is also worth of mentioning that the suggested values k_n^* never exceed 0.8 unlike with the other operators. The average task execution time decreases to about eight seconds within the test period.

Operator OP3 is rather experienced. After the first test round her phase margin generally remains above 63°. As a result, the gain increases constantly. The average task execution times decrease from about eight to less than six seconds during the test period. Operator OP4 had slightly less amount of experience than OP3. Because of knowing the purpose of the experiment, and because after each ten trials, the value of k_n was shown to the operator, OP4 took a goal to rise the value of k_n as high as possible (which was not the actual goal of the experiment). She learned that the SAC algorithm suggests high gains when the operator

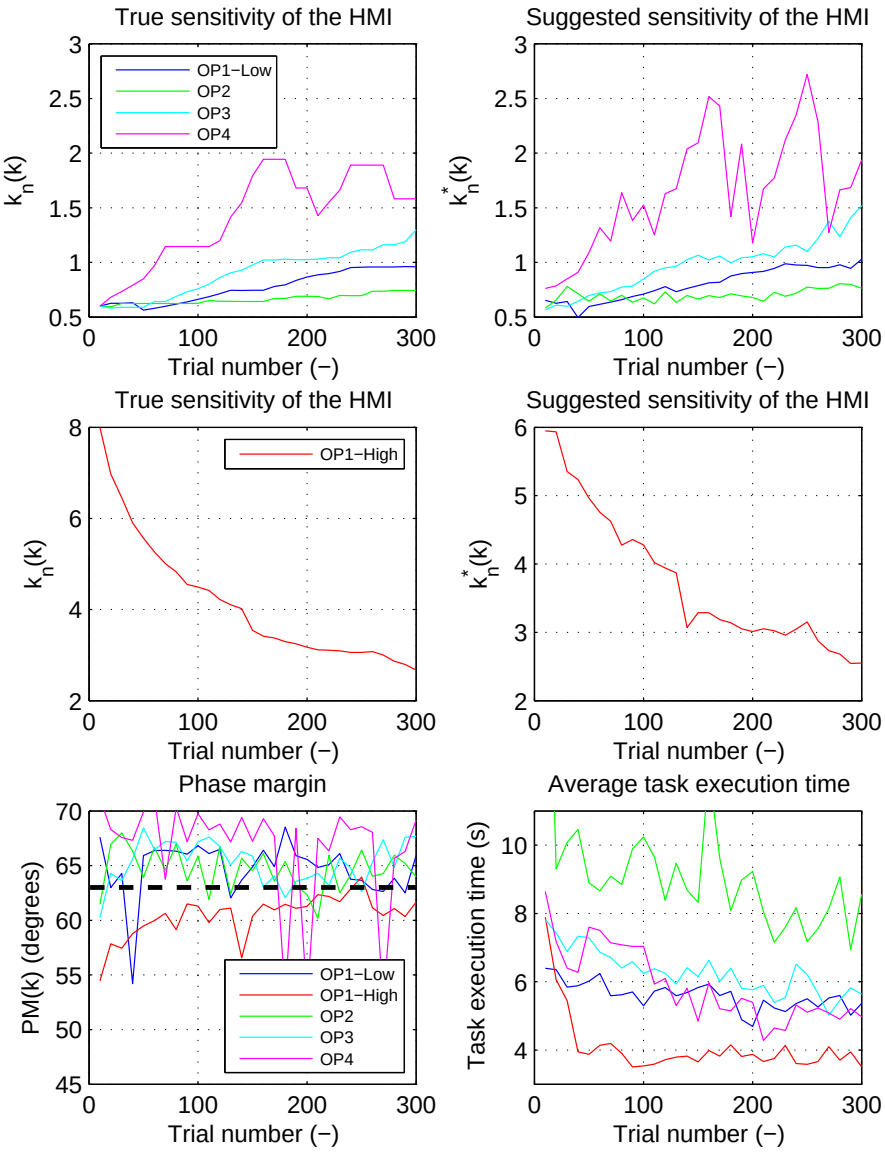


Figure 8.5: The results of the SAC algorithm iteration for the test subjects. The desired phase margin is shown as a dashed thick line on the bottom left plot.

performs full control initial acceleration but thereafter uses only low controls. Thus, she obtained high phase margins. Because she used the full control range in the initial acceleration, the SAC algorithm increased the value of k_n. Nevertheless, despite the high gain values she could retain her strategy and performance, as can be seen from the average task execution time curve. Therefore it is justifiable to state that the values of k_n between $1.5 - 2.0$ are suitable for her skill level.

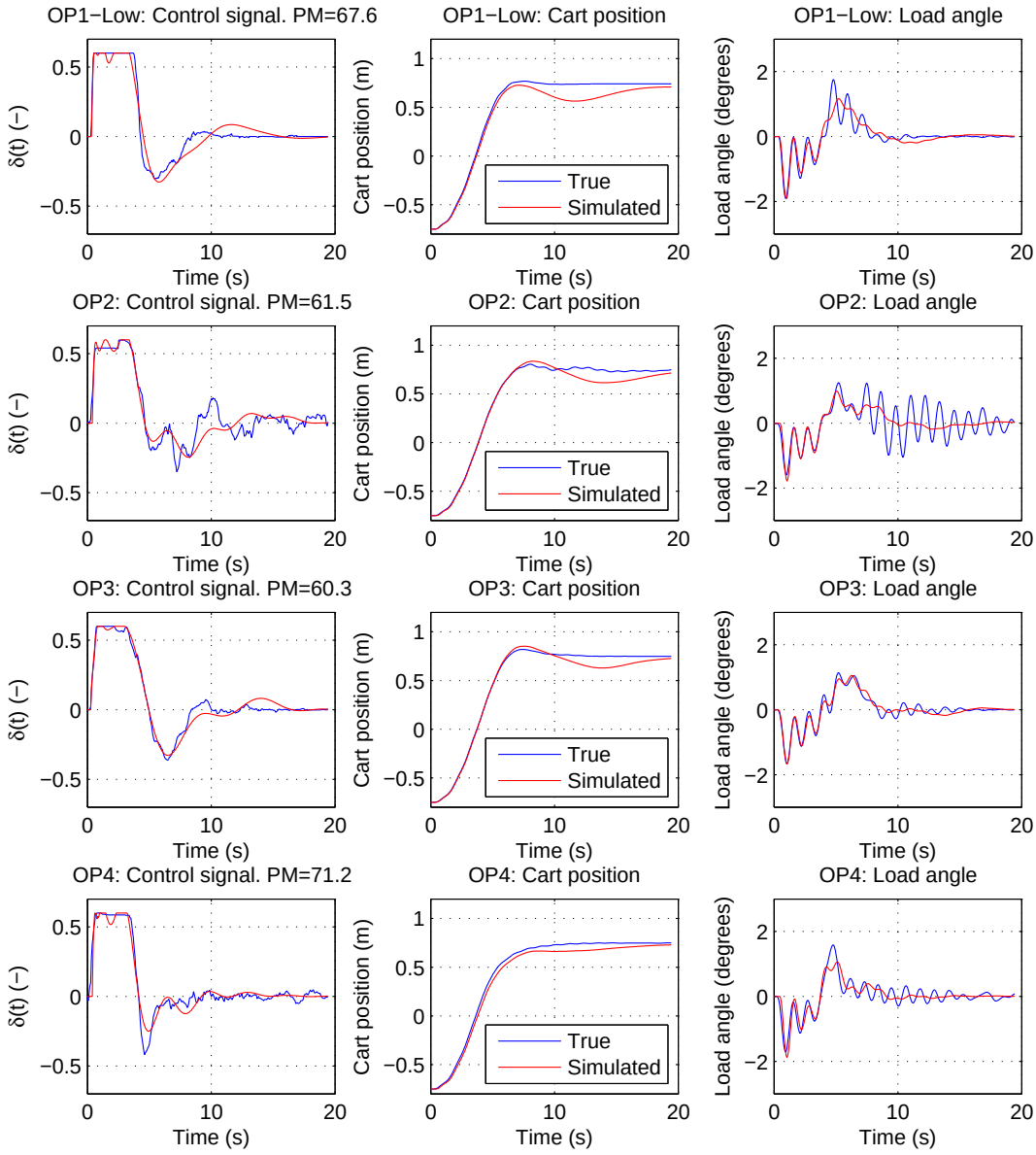


Figure 8.6: Comparison of modeling results for the low initial value cases ($k_n = 0.6$).

To compare the performance of the human modeling for several operators, the modeling results and the corresponding phase margins of the operators OP1-Low, OP2, OP3 and OP4 identified for the first trial round when $k_n = 0.6$ are shown in Figure 8.6. All models manage to capture the first acceleration and deceleration of the trolley. However, the compensation of the angle swinging after deceleration is not modeled well. From the human operator's point of view, the most important reason for poor performance with respect to the angle, is that the angle swinging less than 2° is very difficult to control because it cannot be clearly observed from the

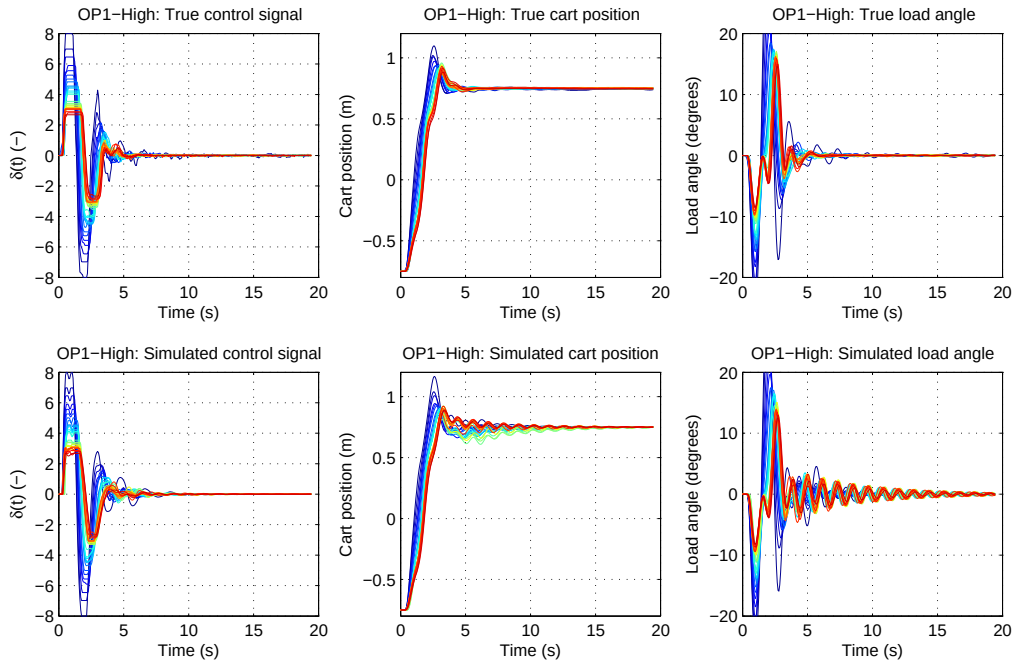


Figure 8.7: Evolution of the operator and system responses during SAC iteration for OP1-High case. The colors change from blue via yellow to red, where red corresponds to the last and best result.

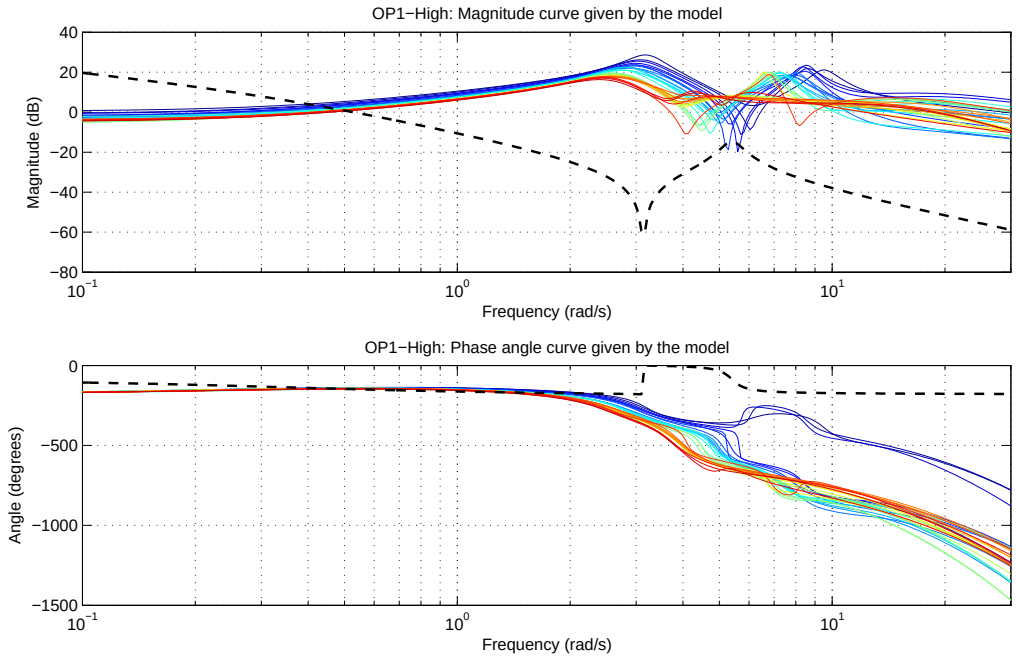


Figure 8.8: Evolution of the operator frequency response during SAC iteration for OP1-High case. The colors change from blue via yellow to red, where red corresponds to the last iteration. The position transfer function is drawn as a thick dashed line.

simulator's screen. In addition, the angle measurement (which would certainly improve the results) is not used as input for the human operator model, because it is not typically available in real-life cranes. Moreover, it should be emphasized that the modeling of the operator is based on only 10 trials of the task. Using more trials for the identification would supress the effect of random control actions typical in the intermittent human operation.

The evolution of the operator OP1-High case responses, the responses of the controlled trolley crane along with the corresponding signals given by the identified human operator model are shown in Figure 8.7. Moreover, the evolution of the frequency response given by the identified model is shown in Figure 8.8. The frequency response of the position transfer function (8.19) of the trolley crane is described as a thick dashed line in Figure 8.8. In both figures, the oldest response (the beginning of the SAC iteration) are described using blue color, and the most recent response (at the end of the experiment) are described using red. The colors in between turn from blue via yellow to red.

Based on the responses, one can conclude that for most of the trial rounds, the corresponding identified model achieves rather good accuracy. The accelerations and decelerations can be modeled very accurately. However, the compensation of the load swinging, especially for small angles cannot be modeled too well. For the gain tuning the result is accurate enough, since the human operator uses the greatest control effort in the position control. It can be seen from the gain curves of the position transfer function and the human operator models that the zero dB cutoff frequency is always less than 2 rad/s, because the position transfer function contains an integrator and thus supresses the high frequencies. Therefore, the inaccuracies in the resonant peaks should not disturb the phase margin calculation. However, the modeling inaccuracies and the impact on the phase margin criterion should be analyzed more closely in the future. Moreover, using different tuning criteria and inclusion of the robust control theory would probably improve the consistency of the tuning in successive trial rounds.

8.7 Experiment: comparison of analytic and knowledge-based methods

Using the analytic approach, one can find HMI parameters which are optimal for the dynamic characteristics of the human operator with respect to control the theoretic design criteria. On the contrary, since the knowledge-based approach is heuristic, it cannot be guaranteed that the resulting parameters are optimal. However, sometimes the knowledge-based method is the only option. Thus, it is interesting to compare whether the results in the SAC experiment in Section 8.6.3 can be repeated using the knowledge-based approach described in Section 8.3.

The first task in the knowledge-based approach is to define the performance (or skill) index. In this case, the overshoot percentage was chosen

$$\mathrm{PO} = 100 \cdot \left(\frac{\max_t z(t)}{z(T)} - 1 \right), \tag{8.26}$$

where $z(t)$ is the cart position and T is the task execution time, because it is related to the phase margin criterion used in the analytic SAC experiment. The overshoot percentage was scaled into a "skill index" by using the statistical learner model approach described in Section 5.4. In practice, this is done by fitting a GEV distribution for the average PO values (average of each ten trials) of the subjects using the maximum likelihood method. The corresponding histogram

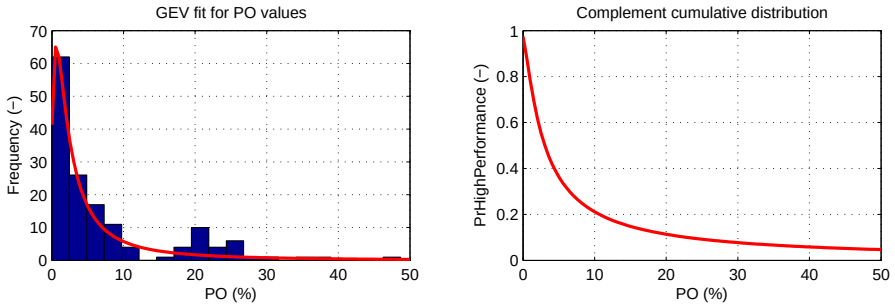


Figure 8.9: The performance/skill evaluation using to the overshoot percentage and GEV.

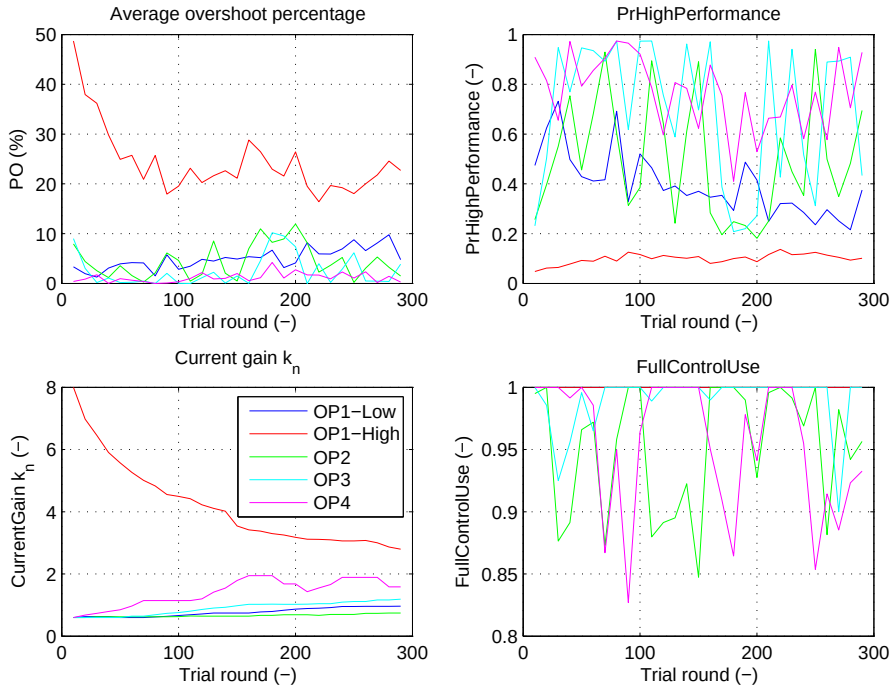


Figure 8.10: The overshoot percentage (PO), the performance evaluation index (PrHighPerformance) according to the scale show in Figure 8.9, the current value of the gain k_n, and FullControlUse.

and the GEV probability density distribution are shown in Figure 8.9 (left). As it can be seen, the fit is not perfect but on the other hand, there were only 150 data points available (30 samples per subject). The right plot in the figure is the complement cumulative distribution (5.11) of the fitted GEV, which is used to obtain the normalized skill index value.

Now the inputs for the parameter adaptation FIS (8.15) can be defined as PrHighPerformance

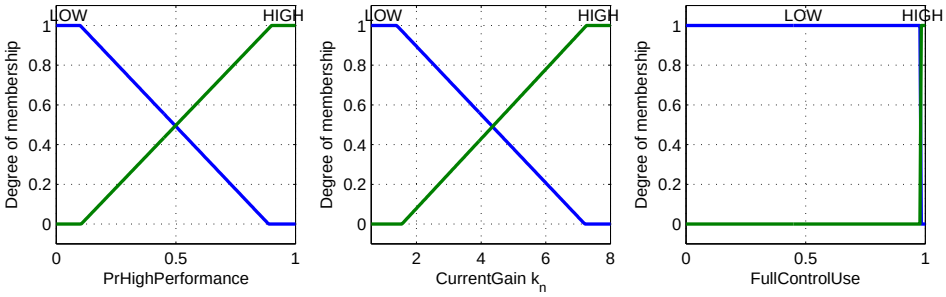


Figure 8.11: The input membership functions of the parameter adaptation FIS.

(Overshoot-based index), CurrentGain $k_n(k)$, and FullControlUse (8.14). The operator-wise values for PO, as well as the corresponding skill index PrHighPerformance, and the other inputs of the parameter adaptation FIS are shown in Figure 8.10. The input membership functions of the FIS are shown in Figure 8.11. Note that the third input is (FullControlIse), in practice, binary-valued. In (8.25), the threshold for the full control use was 0.98. Thus, in order to obtain similar results, the membership functions for FullControlUse are set to switch from LOW to HIGH at 0.98. The rulebase and the structure of the FIS is defined exactly as in Section 8.3. The suggestion with respect to the current gain value are computed using (8.15), with $\alpha = 0.12$.

The suggestions given by the (8.15) were compared to the realized changes of k_n between the trial rounds k and $k+1$ given by the analytic approach in the previous section. The operator-wise comparisons between the approaches are shown in Figure 8.12. The figure also contains the well-known Pearson correlation coefficient R_{pr} between Δk_n values and the Jaccard's similarity measure R_{jc} computed for the signs of the values. The Jaccard's similarity measure is defined by [129]

$$R_{\text{jc}} = \frac{\sum_{k=1}^{N_k} \delta_{\text{sgn}\left(\Delta k_n^{\text{analytic}}(k)\right),\text{sgn}\left(\Delta k_n^{\text{knowledge}}(k)\right)}}{N_k}, \tag{8.27}$$

where N_k is the number of iteration rounds in the experiment (now $N_k = 30$), and $\delta_{i,j}$ is the two-argument Kronecker delta. The idea of studying the Jaccard's similarity measure for the signs is to analyze whether the different approaches provide suggestions to same directions. If all signs are equal $R_{\text{jc}} = 1$. Based on the results it can be seen that the methods provide similar suggestions for OP1-High, OP2, OP3 and OP4 with respect to both R_{pr} and R_{jc}. For OP1-Low, the knowledge-based approach provides constantly suggestions to increase the gain.

8.8 Discussion

The concept of the HMI adaptation system for machine work is proposed. The HMI adaptation is posed as a controller tuning problem, where the human operator model is known (identified), the machine model is known, and the HMI is tuned to be optimal for the current operator dynamics. Three tuning methods are proposed. The choice of the tuning method depends on

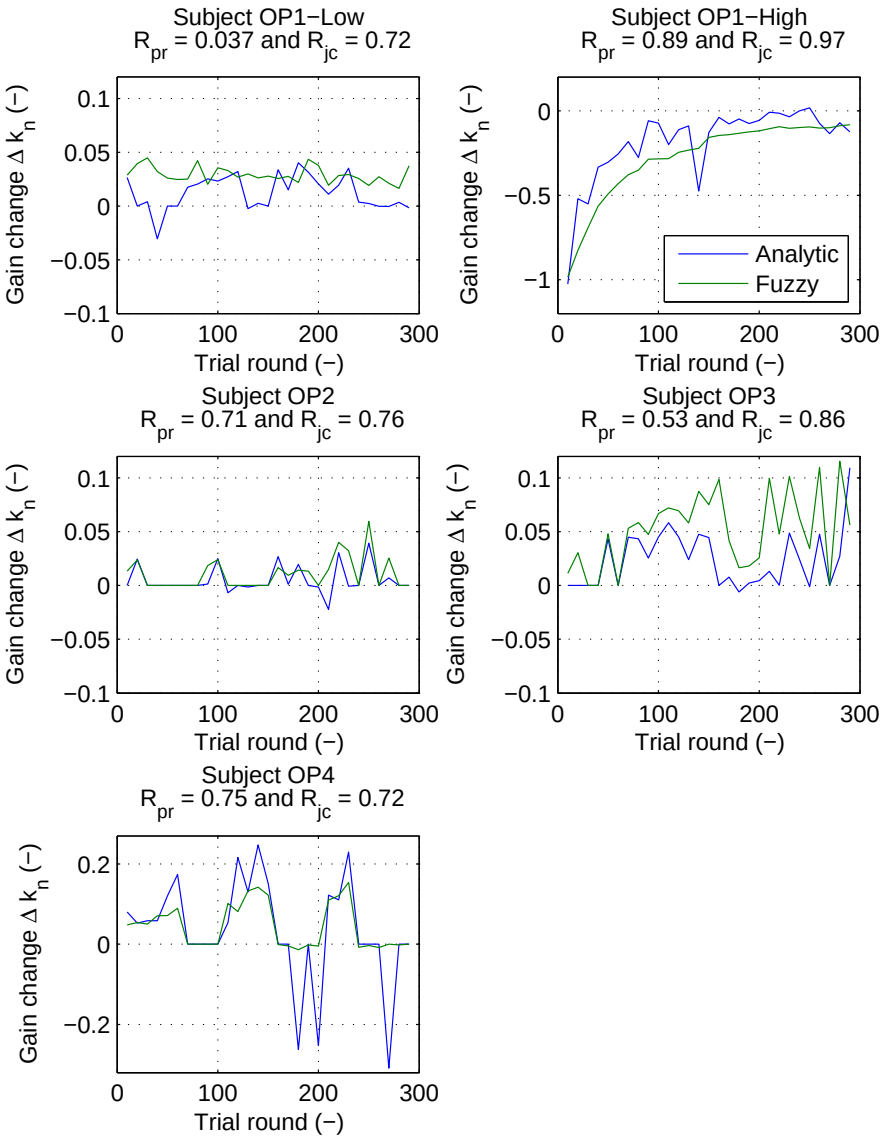


Figure 8.12: Realized change in the HMI sensitivity using the analytic and the knowledge-based approaches.

the used modeling method, as well as on the application. However, if nonlinear models, such as Hammerstein-Wiener type of models are used to model the human operator, one should not use linear control theory based (frequency domain or the optimal tuning) methods, because the Hammerstein-Wiener models lose their accuracy once linearized. Nevertheless, the Hammerstein-Wiener models provide more accurate simulation results and thus using them for the SAC algorithm together with the simulation-based tuning criteria would provide good results. However, the modeling inaccuracies, the nonlinearities, and the tuning criteria have a

strong impact on the result of the HMI adaptation. Therefore, these issues should be studied further in the future.

As an alternative method for the HMI adaptation, the knowledge-based approach is proposed. The method exploits the skill evaluation and the current parameter in an FIS, which performs reasoning whether the current parameters are suitable or not. The knowledge-based approach can be implemented even when the human model is not available.

In order to adapt the system to the human operator's dynamic characteristics, the Skill Adaptive Control (SAC) algorithm is proposed. The algorithm consists on data gathering, modeling, optimization and adaptation steps. Basically the data gathering is done by letting the human operator to perform the task several times, after which the rest of the steps are taken. Then the algorithm starts again from the beginning. The SAC algorithm is implemented in a trolley crane simulator, and an experiment involving several operators is performed. According to the results, the algorithm could tune the system parameters according to the phase margin design criterion. Moreover, the results given by the analytic approach were compared to the knowledge-based approach results. According to the comparison, one can conclude that both approaches can be used to optimize the parameters of the HMI. The analytic method provides results which fulfill a control theoretic design criterion, and thus in that sense, the parameters are optimal. On the other hand, the knowledge-based approach is based only on statistical performance evaluation and heuristic rules. Thus, it cannot be guaranteed that the human operation would fulfill any closed-loop performance criteria after parameter adjustment.

Although the results obtained in the trolley crane experiment are promising, before the implementation of the method in a real-life system, the usability challenges should considered. In general, a significant challenge in the development of the human adaptive systems is to keep the system "familiar" for the human operator. Even the human adaptive system should (at least to some extent) fulfill the basic principles for the direct manipulation systems [54]:

- Predictability
- Overall controllability
- Transparency

Firstly, the behavior of the system should be predictable for the operator. That is, the response of the system should always be same (or it should feel the same for the operator) for the same input. If the system behaves as the human expects it to behave, the predictability requirement is fulfilled. Moreover, from the usability point of view, it should be made possible for the human operator to understand the functioning of the system. That is, the system should be, at least to some extent, transparent. The overall controllability requirement, in this context, means that the user should have the control over the system. He/she should be aware of which functions are left for the user's responsibility. If the HMI parameters are changed, at least the predictability requirement is violated. In practice, the adaptation should be so slow that the human operator is not aware of it. Alternatively, the system should make sure that the human operator understands that the behavior is different every time the HMI is changed. Nevertheless, this thesis is devoted for development of methods to realize the intelligent user interface properties in the control and manipulation of human operated working machines. Thus the usability issues are intentionally left for future research.

9 Summary and conclusions

A new type of Human Adaptive Mechatronics (HAM) system is proposed. The presented Human Adaptive Mechatronics and Coaching (HAMC) includes five interaction levels: *performance assessment*, *skill evaluation*, *coaching*, *assistance (active)*, and *adaptation (passive)*. At the first level of interaction, that is, the performance assessment, the system evaluates the performance in the production level. The operator can see the performance on the screen of the machine. This is typically the situation in the working machines nowadays. At the second level of interaction, the system evaluates the operator's performance and skill level, and shows the performance overall for the operator. For a motivated operator, such a feedback can already be enough to improve one's performance. At the coaching level, the system can give detailed feedback and suggestions in linguistic form concerning separately defined coaching problems (tasks the operator performs with the machine). The coaching exploits the paradigm of Intelligent Coaching Systems (ICSs) combined with skill evaluation results achieved in different areas such as robotics, telesurgery, and mobile working machines. At the active assistance level, the system can automatically intervene with the control commands of the human operator. For example, when the human operator is using an overhead crane, the system can automatically take care of the load swing control so that the maneuverability would be easier. On the contrary, in the passive adaptation, the system evaluates the human skill or models the operator's characteristics and then uses the result to find the optimal HMI parameter configuration for the current characteristics of the human operator.

The different interaction levels require solutions for the following subproblems: *intent recognition*, *skill evaluation/human modeling*, *coaching*, *adaptation* and *assist control law design*. This thesis focused on the intent recognition, the skill evaluation, the human modeling, and the adaptation. The intent recognition is carried out using the HMM-based work cycle modeling method, where the work of the operator is divided into tasks, which can be recognized using the existing measurement information and the trained HMM. In this thesis, the method is implemented in three industrial applications. The intent recognition provides the information about what the operator is trying to do at a given time. This information is then used in the skill evaluation, which is based on skill indices derived using the work cycle recognition, as well as the concept of the statistical learner model. In this thesis, the skill evaluation method is implemented in three industrial applications. The human operator modeling is also an important problem in the area of HAM. In this thesis the modeling is performed using the structural models, which are based on the model of the controlled machine.

The structural models such as MOCM retain the mapping from the parameters of the controlled machine to the parameters of the human operator model. Thus, at least in a small range, the model can be made adaptive to the changes in the machine dynamics. This thesis proposes a Hammerstein-Wiener extension to the existing Modified Optimal Control Model (MOCM) structure to account for the nonlinearities, for example, in the HMI. Moreover, a batch type parameter estimation method for HWMOCM is developed. In addition to the MOCM, the Data-Based State Feedback Model (DBSFM) is proposed. For the DBSFM any optimal control type of model structure can be used. In the DBSFM the optimality assumptions are relaxed so that gradient-based parameter estimation is possible. In addition, it is proposed that the optimality of the model can be preserved using the return difference condition constraints.

Moreover, it is shown how the DBSFM can be returned to the optimal control framework using the inverse optimal control solution. The Linear Matrix Inequality (LMI)-based solution for the inverse optimal control in terms of DBSFM is presented.

The discrete version of the DBSFM (DDBSFM) is presented. The derivation of the control gain and estimation parameters of the DDBSFM is done in the discrete time domain. The suboptimal inverse optimal control solution is presented to convert the DDBSFM to the optimal control framework. Only the suboptimal inverse solution could be found because in the literature the criteria for the inverse optimality are not as simple in discrete time as in continuous time. The gradient-based batch identification methods for the DBSFM and the DDBSFM are proposed. In addition, the Hammerstein-Wiener structure for the DDBSFM is proposed, and the corresponding gradients are derived for the parameter estimation method. The identification methods are implemented for modeling a human operator controlling a trolley crane simulator in two cases: linear and nonlinear HMI. In addition to the structural models, the polynomial models Autoregressive with Exogenous Input (ARX) and Hammerstein-Wiener Output Error Model (HWOE) are identified. In all cases, the Hammerstein-Wiener models performed better than their linear counterparts. The performance of DBSFM and DDBSFM was comparable for the HWOE and ARX models of same order. However, the DBSFM and the DDBSFM have less free parameters, and they retain the mapping from the machine parameters to the human operator model parameters. The return difference criteria for the DBSFM are studied and the inverse optimal control solution is determined. The suboptimal inverse solution is determined for the DDBSFM.

The passive adaptation interaction level is studied. The concept of human skill adaptive HMI is proposed to emphasize the system's ability to adapt to the skill level and the dynamic characteristics of an individual operator. The passive adaptation property is posed as a controller tuning problem, where the machine model is known and the human operator model can be identified (i.e. known). By using the machine and the human model, the HMI can be designed to be optimal for the current characteristics of the human operator. Three different tuning methods for finding the optimal HMI parameters are proposed: the simulation-based tuning, the frequency domain tuning, and the Lyapunov-based optimal tuning. The results of the tuning methods are compared using the MOCM identified in the human operator modeling experiment. As an alternative method for the HMI adaptation, the knowledge-based approach is described. The approach exploits the skill evaluation and Fuzzy Inference Systems (FISs). It can be used even when the human operator model is not available.

Because the human adapts oneself to changes in the HMI, the SAC adaptation algorithm is proposed. The SAC algorithm adapts the system to the human operator's dynamic characteristics little-by-little so that eventually the HMI parameters which are optimal for the human operator are found. In addition, while the human operator is learning, the algorithm changes the HMI to better suit for the current level of the skill. The SAC algorithm is implemented in a trolley crane simulator, and the tests for four subjects are performed. According to the results, the algorithm could find the suitable parameters for the human operators, with respect to the phase margin tuning criterion.

The development of the future intelligent machines requires that the machine can understand the human operator. That is, the machine can recognize and predict the operator's intent and plan and evaluate the performance and the skill. Thus, the machine can adapt itself to

optimize the usability, the maneuverability and the cooperation between the human and the machine *individually* for the current operator. The most important difference between the traditional human-machine system design and the HAM approach studied in this thesis is that in the conventional systems, the design, such as the HMI and the possible assisting control are designed to be same for every operator. In the HAM approach the design is done so that it is optimal for an individual operator. That is, based on the skill evaluation results or the human operator model, the system can change itself to be optimally designed for the individual operator. This is important, since the difference between a beginner and an expert can be significant, in terms of the control characteristics and the skill level. The expert can handle high sensitivity of the HMI and thus achieve good task performance, whereas the beginner, given the optimal sensitivity of the HMI that the expert uses, obtains poor task performance, because the movements are too fast for the beginner's skill level.

The *coaching* of the human operators according to the skill evaluation results plays an important role when aiming at improving the operators' work performance. The skill evaluation methods described in this thesis provide a solid base to implement the coaching system. Although this thesis does not focus on coaching, already the HMI parameter optimization results (suggestions given by the analytic or knowledge-based approach) can be shown for the operator as instructions, if not adapted automatically. This is also a form of coaching in the *parameter tuning* coaching problem.

The skill evaluation is a basis for an ICS. In comparison with the traditional simulator-based ICSs, the implementation of the methods in the real-world applications to evaluate the skills of the operator performing multiobjective and demanding work with a mobile working machine is challenging. Yet another challenge is that in the work environment, the most important variables measurable in the simulator environment, are immeasurable. The first problem in the skill evaluation is to recognize the operator's intent based on the existing measurements. That is, what the operator is trying to do? Based on the author's experience in several industrial case studies, the HMM-based work cycle modeling method is shown to be a powerful tool for the intent recognition. For the skill evaluation, it is enough that the intent can be recognized after the task is executed. Thus, at this point, the online intent recognition is not needed. Once the intent has been recognized, that is, it is known what the operator is trying to do, the next task is to evaluate how well he/she is doing the work. In this thesis, the skill evaluation is carried out using skill indices derived based on the measurements and the work cycle recognition results. The uncertainty inherent in the human operator action is compensated using the statistical learner model. According to the experimental results, as well as to the evaluations given by the domain experts (in the forest machine case), the skill evaluation approach proposed in this thesis provide reliable results. The reliability of the results can be increased even more by developing better measurements to account for the operating conditions variations.

According to the human operator modeling results, the structural models managed to perform comparably to the simpler polynomial models in the two modeling cases. The return difference condition constraints could guarantee the existence of the inverse optimal control solution. However, the inclusion of the return difference constraints slows down the parameter estimation algorithm. In this thesis, only batch type of estimation algorithms are studied, because for the passive adaptation, it is enough to have a human operator model after the task has been performed. Thus, for example modeling a human operator controlling a crane, the model can

be identified based on several executions of a task, rather than using the online estimation methods. In the literature, there are claims that linear models cannot be used to model the human operator reliably. However, according to the results of this thesis, the linear models are sufficient, if the controlled machine itself is not strongly nonlinear. The nonlinearities in the HMI and the measurement end of the human operator model can taken into account using the Hammerstein-Wiener structure. However, as it was seen from the frequency responses of the linearized Hammerstein-Wiener models, the frequency responses of the linear parts are drastically different from the frequency responses of the linear models. Therefore, one should be careful when using the Hammerstein-Wiener models. They provide excellent simulation results, since they can account for the human operator's behavior under nonlinearities in the input and the output. However, in general, the Hammerstein-Wiener models should not be used without the nonlinearities, as it is the case in using any nonlinear models.

The current parameter estimation algorithms of the structural models are not very fast compared to the simplest models, especially if the controlled machine is of high order. Therefore, the author's opinion is that if the adaptability to the operating point is not needed, one should use the simpler polynomial-type models such as ARX or HWOE. The DBSFM/DDBSFM are promising methods but there are still development issues regarding the parameter estimation and the missing feedforward structure.

By posing the HMI adaptation as a controller tuning problem, familiar methods from the control theory can be used to realize the passive adaptation property. In addition, the optimal control models' stochastic cost functions can be expressed theoretically. Thus, for these models, a new theoretical tuning method is proposed. Moreover, even when the current measurement capability or the complexity of the work does not allow the human operator modeling, the knowledge-based approach can be used to adapt the HMI parameters. According to the experimental results, the current formulation of the SAC algorithm performs correctly if the human operator's control strategy stays constant. For practical applications, the adaptivity should be so slow that the human operator would not perceive the change in the system's behavior. In addition, the occasional changes in the human operator's control strategy would not harm the parameter adaptation. In this way, the principles of direct manipulation systems would not be violated.

The proposed intent recognition and skill evaluation methods which are implemented in several industrial applications are very promising results in the way of the development of intelligent human adaptive cooperative and coaching machines. The methods are extendable for several applications. The proposed methods have the greatest potential in machines where the human operator is continuously in charge, because, the operator's work tasks and the operating conditions are typically varying, and the some of the important signals cannot typically be measured. The human operator modeling and HMI adaptation results obtained in this thesis are promising steps towards online modeling e.g. of the human operator in real-world industrial cranes. All in all, the methods proposed in this thesis improve the machine's understanding of the human operator so that it can try it's everything to make the overall system to perform better, to improve the human operator's proficiency, to improve the usability of the machine, and to improve the productivity, fuel economy and safety of the work. In addition to the machine work, there are already results showing that in the process industry, the human skill can be evaluated and the outcome and profitability of the process could be increased by assisting

the operators in the process control.

Although this thesis applies the results to working machines, the possible fields of applications are significantly wider. In many western countries the number of elderly people in the population has become disproportional to the number of working-aged people. The intelligent human adaptive mechatronic devices could improve the life quality of the elderly and disabled people by providing psysical, social, and mental assistance. Moreover, the skill evaluation methods as well as the assistive and human adaptive technologies could help in training of human operators to perform the work efficiently and safely in fields such as commercial and military aeroplances, automotive industry, and car driving. The safety is also a critical issue in the medical field, especially in surgery. The teleoperated surgery devices could exploit the human adaptive technologies during normal operation. In addition, the training of new surgeons could be made faster and more efficient by utilizing the skill evaluation and intelligent coaching methods to continuously assess and coach the surgeon trainees.

Personal and professional exercise and sport field has numerous potential applications for skill evaluation, and human adaptive technologies. A player would get instant feedback about the possible development issues in the execution of the exercise. Such a personal trainer system would continuously push the player towards better performance, but simultaneously would ensure the ergonomy of the training and thus minimize the risk of injury. The improvement in the exercise techniques and motivation brought by the assessment and coaching would increase the health and well-being of normal people. Similar advantages could be achieved in normal work life environments. The ergonomy of the work, e.g. physical warehouse work could be increased by evaluation of the workers' lifting techniques and working positions and thus coaching towards better ergonomy. As a result, the joint, back and muscle injuries would be decreased, which would increase the life quality of the workers and reduce the costs spent for rehabilitation and treatment of injuries.

In conclusion, let us consider the slogan "why should people adapt to systems, systems should adapt to people instead?" again. Definitely, in the future the system should adapt to people, because there is a significant amount of improvement potential with respect to productivity, energy efficiency, safety and ergonomy due to the poor human-machine cooperation, poor machine tuning, and insufficient human skills. In addition, *it is possible.*

References

[1] D. Aarno, S. Ekvall, and D. Kragic. 2005. Adaptive Virtual Fixtures for Machine-Assisted Teleoperation Tasks. In: Proceedings of the IEEE International Conference on Robotics and Automation, pages 1139–1144.

[2] D. Aarno and D. Kragic. 2006. Layered HMM for motion intention recognition. In: Proceedings of the IEEE/RSJ Int. Conf. Intelligent Robots and Syst., pages 5130–5135.

[3] J. J. Abbott, G. D. Hager, and A. M. Okamura. 2003. Steady-hand teleoperation with virtual fixtures. In: Proceedings of the 12th IEEE International Workshop on Robot and Human Interactive Communication, pages 145–151.

[4] A. Abdel-Malek and V. Z. Marmarelis. 1988. Modeling of Task-Dependent Characteristics of Human Operator Dynamics Pursuit Manual Tracking. IEEE Transactions on Systems, Man, and Cybernetics 18, pages 163–172.

[5] Bruce Abernethy. 2005. The biophysical foundations of human movement. Human Kinetics.

[6] V. Aleven, K. Ashley, C. Lynch, and N. Pinkwart (eds). 2008. Intelligent Tutoring Systems for Ill-defined domains: Assessment and Feedback in Ill-Defined Domains. In: Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains ITS 2008. Montreal, Canada.

[7] M. Alexík. 2000. Modelling and identification of eye-hand dynamics. Simulation Practice and Theory 8, pages 25–38.

[8] B. D. O. Anderson and J. B. Moore. 1990. Optimal Control: Linear Quadratic Methods. Prentice-Hall International Inc., Englewood Cliffs, NJ, USA.

[9] J. R. Anderson. 1991. Cognitive skills and their aquisition. Erlbaum, Hillsdale, NJ, USA.

[10] M. R. Anderson. 1994. A Model of the Human Operator Using Sensitivity Function Shaping. In: Proceedings of the American Control Conference. Baltimore, Maryland, USA.

[11] Anon. 2000. Flight Control Design – Best Practices. Technical report, North Atlantic Treaty Organization.

[12] Anon. 2001. Timbermatic^{TM} 300 Control and Measurement system (in Finnish). Tampere, Finland.

[13] Anon. 2010. Optimization Toolbox 5.0. URL `http://www.mathworks.com /products/optimization/`.

[14] F. Arai, M. Yoneda, T. Fukuda, K. Miyata, and T. Naito. 1995. Operational assistance of the crane system by the interactive adaptation interface. In: Proceedings of the 4th IEEE International Workshop on Robot and Human Communication, pages 333–338. Tokyo, Japan.

[15] M. Arif and H. Inooka. 1999. Iterative manual control model of human operator. Biological Cybernetics 81, pages 445–455.

[16] H. Asada and S. Liu. 1991. Transfer of human skills to neural net robot controllers. In: Proceedings of the IEEE International Conference on Robotics and Automation, volume 3, pages 2442–2448.

[17] S. Aulanko and K. Tervo. 2009. Modeling and evaluation of harbor crane work. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC2009), pages 869–874. San Antonio, USA.

[18] S. Aulanko and K. Tervo. 2010. Modeling and analysis of harbor crane work efficiency using work cycle recognition. In: Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Montreal, Canada.

[19] S. Baron, D. L. Kleinman, and W. H. Levison. 1970. An Optimal Control Model of Human Response - Part II: Prediction of Human Performance in a Complex Task. Automatica 6, no. 3, pages 371–383.

[20] Charlie Bass. 2009. Demonstrating the future of lifting technology. Way Up – A Magazine about Lifting Businesses (Konecranes Magazine) , no. 2.

[21] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager. 2004. Vision-assisted control for manipulation using virtual fixtures. IEEE Transactions on Robotics 20, no. 6, pages 953–966.

[22] Aude G. Billard, Sylvain Calinon, and Florent Guenter. 2006. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. Robotics and Autonomous Systems 54, no. 5, pages 370–384. The Social Mechanisms of Robot Programming from Demonstration.

[23] E.R. Boer and R.V. Kenyon. 1998. Estimation of time-varying delay time in nonstationary linear systems: an approach to monitor human operator adaptation in manual tracking tasks. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 28, no. 1, pages 89–99.

[24] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. 1994. Linear matrix inequalities in system and control theory. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, USA.

[25] J. Breuker. 1998. What are Intelligent Coaching Systems and Why are they (in)evitable? In: Proceedings of the IEE Colloquium on Artificial Intelligence in Educational Software, pages 2/1–2/5. London, UK.

[26] Carey Bunks and Dan McCarthy. 2000. Condition-based mainenance of machines using hidden Markov models. Mechanical Systems and Signal Processing 14, pages 597–612.

[27] Juan L. Cabrera and John G. Milton. 2001. On-Off Intermittency in a Human Balancing Task. Physical Review Letters 89, no. 15, pages 158702–158706.

[28] Juan L. Cabrera and John G. Milton. 2003. Stick balancing: On-off intermittency and survival times. Nonlinear Stud. 10, no. 2, pages 1–13.

[29] Liang-Kuang Chen. 2001. Identification of a Driver Steering Model, and Model Uncertainty, From Driving Simulator Data. Journal of Dynamic Systems, Measurement, and Control 123, pages 623–629.

[30] Liang-Kuang Chen and A. Galip Ulsoy. 2000. Identification of a Nonlinear Driving Model via NARMAX Modeling. In: Proceedings of the American Control Conference. Chicago, Illinois.

[31] J. B. Davidson and D. K. Schmidt. 1994. Extended Cooperative Control Synthesis. Technical Memorandum L-17212, NASA, Langley Research Center Hampton, VA, USA.

[32] John B. Davidson and David K. Smith. 1992. Modified Optimal Control Pilot Model for Computer-Aided Design and Analysis. Technical Memorandum L-16979, NASA, Langley Research Center Hampton, VA, USA.

[33] I. I. Delice and S. Ertugrul. 2007. Intelligent Modeling of Human Driver: A Survey. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pages 648–651. Istanbul, Turkey.

[34] David B. Doman. 1998. Projection methods for order reduction of optimal human operator models. Ph.D. thesis, Virginia State University, VA, USA.

[35] David B. Doman and Mark R. Anderson. 2000. A Fixed-order optimal control model of human operator response. Automatica 36, pages 409–418.

[36] S. Ekvall, D. Aarno, and D. Kragic. 2006. Online task recognition and real-time adaptive assistance for computer-aided machine control. IEEE Transactions on Robotics 22, no. 5, pages 1029–1033.

[37] Seniz Ertugrul. 2008. Predictive modeling of human operators using parametric and neuro-fuzzy models by means of computer-based identification experiment. Engineering Applications of Artificial Intelligence 21, no. 2, pages 259–268.

[38] Janet Finlay and Russell Beale. 1993. Neural networks and pattern recognition in human-computer interaction. SIGCHI Bull. 25, no. 2, pages 25–35.

[39] P. M. Fitts and J. R. Peterson. 1964. Information capacity of discrete motor responses. Journal of Experimental Psychology 67, no. 2, pages 103–112.

[40] M. Friman and H. Happonen. 2007. Managing adaptive process monitoring: New tools and case examples. In: Proceedings of the IEEE Mediterranean Conference on Control and Automation, pages 27–29.

[41] K. Furuta, M. Iwase, and S. Hatakeyama. 2005. Internal Model and Saturating Actuator in Human Operation from View of Human-Adaptive Mechatronics. IEEE Transactions on Industrial Electronics 52, no. 5, pages 1236–1245.

[42] K. Furuta, Y. Kado, and S. Shiratori. 2006. Assisting control in Human Adaptive Mechatronics – single ball juggling. In: Proceedings of the IEEE International Conference on Control Applications, pages 545–550.

[43] K. Furuta, Y. Kado, and S. Shiratori. 2007. Identification of dynamic characteristics in human operation. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pages 2977–2982.

[44] S. Gellerstedt. 2002. Operation of the single-grip harvester: motor-sensory and cognitive work. International Journal of Forest Engineering 13, no. 2, pages 35–47.

[45] J.D. Gibbons and S. Chakraborti. 2003. Nonparametric statistical inference. Statistics, textbooks and monographs. Marcel Dekker. URL `http://books.google.fi/books?id=kJbVO2G6VicC`.

[46] Avelino J. Gonzalez, Ronald F. DeMara, William J. Gerber, and Michael Georgiopoulos. 2004. Context-driven near-term intention recognition. The Journal of Defence Modeling and Simulation: Applications, Methodology, Technology 1, no. 3, pages 153–170.

[47] F. Harashima. 2005. Human Adaptive Mechatronics – Interaction and Intelligence. In: Proceedings of the 10th IEEE Conference on Emerging Technologies and Factory Automation.

[48] F. Harashima and S. Suzuki. 2006. Human adaptive mechatronics - interaction and intelligence. In: Proceedings of the IEEE International Workshop on Advanced Motion Control, pages 1–8.

[49] T. J. Harris and C. T. Seppala abd L. D. Desborough. 1999. A review of performance monitoring and assessment techniques for univariate and multivariate control systems. Journal of Process Control 9.

[50] S. Hatakeyama, M. Iwase, and S. Yamaura. 2005. Analysis of Human Operation Utilizing Closed-loop Identification Method. In: Proceedings of the IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pages 232–236.

[51] C. Heinze. 2003. Modeling Intention Recognition for Intelligent Agent Systems. Ph.D. thesis, University of Melbourne, Australia.

[52] K. Hidaka, K. Saida, and S. Suzuki. 2006. Relation between skill level and input-output time delay. In: Proceedings of the IEEE International Conference on Control Applications, pages 557–561.

[53] Omar B. Hijab. 1983. The Adaptive LQG Problem – Part I. IEEE Transactions on Automatic Control 28, no. 2.

[54] Kristina Höök. 2000. Steps to take before Intelligent User Interfaces become real. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi= 10.1.1.28.5093`.

[55] Vesa Hölttä. 2004. Analysis of Stem Diameter Measurement In Forest Harvester. Master's thesis, Helsinki University of Technology, Control Engineering Laboratory, Helsinki, Finland.

[56] Vesa Hölttä. 2009. Plant performance evaluation in complex industrial applications. Ph.D. thesis, Helsinki University of Technology, Espoo, Finland.

[57] Vesa Hölttä, Matti Repo, Lauri Palmroth, and Aki Putkonen. 2005. Index-based performance assessment and condition monitoring of a mobile working machine. In: Proceedings of IDETC'05 2005 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Long Beach, California, USA.

[58] Andreas G. Hofmann and Brian C. Williams. 2007. Intent Recognition for Human-Robot Interaction. In: Proceedings of the AAAI Spring Symposium: Interaction Challenges for Artificial Assistance, volume SS-07-04, pages 60–61.

[59] http://global.wonderware.com/en/pages/default.aspx (Cited at April 20 2010). Wonderware website.

[60] http://www.plctutor.com/ (Cited at April 20 2010). PLCTutor website.

[61] S. Iba, C.J.J. Paredis, and P.K. Khosla. 2003. Intention aware interactive multi-modal robot programming. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, volume 4, pages 3479–3484.

[62] Hiroshi Igarashi. 2008. Subliminal Calibration for Machine Operation Systems. In: Proceedings of the International Conference on Control, Automation and Systems, pages 2812–2817. COEX, Seoul, Korea.

[63] Hiroshi Igarashi. 2009. Subliminal calibration for machine operation. In: Proceedings of the 35th IEEE Annual Conference on Industrial Electronics, pages 4268–4273.

[64] A. Imura, T. Takagi, and T. Yamaguchi. 1993. Intention recognition using conceptual fuzzy sets. In: Proceedings of the Second IEEE International Conference on Fuzzy Systems, volume 2, pages 762–767.

[65] Y. Inagaki, H. Sugie, H. Aisu, S. Ono, and T. Unemi. 1995. Behavior-based intention inference for intelligent robots cooperating with human. In: Proceedings of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium, volume 3, pages 1695–1700.

[66] R. Isermann. 2006. Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer.

[67] M. Ito. 2000. Internal Model Visualized. Nature 403, pages 153–154.

[68] E. Itoh and S. Suzuki. 2005. Nonlinear Approach for Human Internal Models: Feedforward and Feedback Roles in Pilot Maneuver. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, volume 3, pages 2455–2462.

[69] Y. Iwai, M. Iwase, R. Ohno, and S. Hatakeyama. 2006. Identification of Human Operation under Closed-loop System. In: Proceedings of the 32nd IEEE Annual Conference on Industrial Electronics, pages 103–107.

[70] M. Iwase, S. Hatakeyama, and K. Furuta. 2006. Analysis of intermittent control systems by human operations. In: Proceedings of the 32nd IEEE Annual Conference on Industrial Electronics, pages 4516–4521.

[71] G. Johannsen. 1972. Development and Optimization of a Nonlinear Multiparameter Human Operator Model. IEEE Transactions on Systems, Man, and Cybernetics 2, no. 4, pages 494–504.

[72] George A. Baker Jr. and Peter Graves-Morris. 1996. Padé Approximants. Cambridge University Press, 2 edition.

[73] R. Kalman. 1964. When is a linear control system optimal? ASME Transactions, Journal of Basic Engineering 86, pages 51–60.

[74] A. Kandel. 1991. Fuzzy expert systems. CRC Press.

[75] S. Katsura and K. Ohnishi. 2004. Human cooperative wheelchair for haptic interaction based on dual compliance control. IEEE Transactions on Industrial Electronics 51, no. 1, pages 221–228.

[76] Mitsuo Kawato. 1999. Internal models for motor control and trajectory planning. Current Opinion in Neurobiology 9, no. 6, pages 718–727.

[77] Jung-Hyun Kim, Dong-Gyu Kim, Jeong-Hoon Shin, Sang-Won Lee, and Kwang-Seok Hong. 2005. Hand Gesture Recognition System Using Fuzzy Algorithm and RDBMS for Post PC, volume 3614 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 170–175 pages.

[78] Sa Kil Kim, Je Yun Park, and Seong Nam Byun. 2009. Crew resource management training for improving team performance of operators in Korean advanced nuclear power plant. In: IEEE International Conference on Industrial Engineering and Engineering Management, pages 2055–2059.

[79] G. Klein. 1998. Sources of power: How people make decisions. MIT Press, Cambridge, MA, USA.

[80] D. L. Kleinman, S. Baron, and W. H. Levison. 1970. An Optimal Control Model of Human Response - Part I: Theory and Validation. Automatica 6, no. 3, pages 357–369.

[81] D. L. Kleinman, S. Baron, and W. H. Levison. 1971. A Control Theoretic Approach to Manned-Vehicle Systems Analysis. IEEE Transactions on Automatic Control 16, pages 824–832.

[82] A. J. Koivo and D. W. Repperger. 1997. Skill evaluation of human operators. In: IEEE International Conference on Systems, Man, and Cybernetics, volume 3, pages 2103–2108.

[83] Heikki N. Koivo. 1974. Least-Squares Estimator for Hereditary Systems with Time-Varying Delay. IEEE Transactions on Systems, Man, and Cybernetics 4, no. 3, pages 276–283.

[84] K. Kosuge, K. Furuta, and T. Yokoyama. 1987. Virtual internal model following control of robot arms. In: Proceedings of the IEEE International Conference on Robotics and Automation, volume 4, pages 1549–1554.

[85] S. Kotz and S. Nadarajah. 2000. Extreme Value Distributions: Theory and Applications. World Scientific.

[86] P. Krauthausen and U.D. Hanebeck. 2009. Intention recognition for partial-order plans using Dynamic Bayesian Networks. In: Proceedings of the 12th International Conference on Information Fusion, pages 444–451.

[87] D. Kulic, Dongheui Lee, C. Ott, and Y. Nakamura. 2008. Incremental learning of full body motion primitives for humanoid robots. In: Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots, pages 326–332.

[88] Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. 2008. Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains. Int. J. Rob. Res. 27, no. 7, pages 761–784.

[89] K. Kurihara, S. Suzuki, F. Harashima, and K. Furuta. 2004. Human adaptive mechatronics (HAM) for haptic system. In: Proceedings of the 30th IEEE Annual Conference on Industrial Electronics.

[90] Keiichi Kurihara, Satoshi Suzuki, and Katsuhisa Furuta. 2006. Elucidation of skilled human controller on stabilization with voluntary motion. In: Proceedings of the IEEE International Conference on Control Applications and the IEEE International Symposium on Intelligent Control, pages 573–578.

[91] Y. Labit, D. Peaucelle, and D. Henrion. 2002. SeDuMi Interface 1.02 : a Tool for Solving LMI Problems with SeDuMi. In: Proceedings of the CACSD Conference. Glasgow.

[92] Vladimir S. Larin. 2003. About the inverse problem of optimal control. Applied and Computational Mathematics 2, no. 2, pages 90–97.

[93] Paul H. Lewis and Chang Yang. 1997. Basic Control Systems Engineering. Prentice-Hall.

[94] J. Löfberg. 2004. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In: Proceedings of the CACSD Conference. Taipei, Taiwan. URL `http://control.ee.ethz.ch/ joloef/yalmip.php`.

[95] Lennart Ljung. 1998. System Identification: Theory for the User (2nd Edition). Prentice Hall PTR.

[96] Maurice R. Masliah and Paul Milgram. 2000. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In: CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 25–32. ACM, New York, NY, USA.

[97] H. Mayer, D. Burschka, A. Knoll, E.U. Braun, R. Lange, and R. Bauernschmitt. 2008. Human-machine skill transfer extended by a scaffolding framework. In: IEEE International Conference on Robotics and Automation, pages 2866–2871.

[98] D. T. McRuer. 1995. Pilot-Induced Oscillations and Human Dynamic Behaviour. Technical report, NASA, CR 4683.

[99] D.T. McRuer and H.J. Jex. 1967. A Review of Quasi-Linear Pilot Models. IEEE Transactions on Human Factors in Electronics 8, pages 231–249.

[100] G. Megali, S. Sinigaglia, O. Tonet, and P. Dario. 2006. Modelling and Evaluation of Surgical Performance Using Hidden Markov Models. Biomedical Engineering, IEEE Transactions on 53, no. 10, pages 1911–1919.

[101] W. Mendenhall, R.J. Beaver, and B.M. Beaver. 2009. Introduction to probability and statistics. Brooks/Cole, Cengage Learning.

[102] C. Mitsantisuk, S. Katsura, and K. Ohishi. 2009. Kalman-Filter-Based Sensor Integration of Variable Power Assist Control Based on Human Stiffness Estimation. Industrial Electronics, IEEE Transactions on 56, no. 10, pages 3897–3905.

[103] J. Miyata, Y. Kaida, and T. Murakami. 2008. $v-\dot{\phi}$-Coordinate-Based Power-Assist Control of Electric Wheelchair for a Caregiver. IEEE Transactions on Industrial Electronics 55, no. 6, pages 2517–2524.

[104] Y. Morita, H. Marumo, M. Uchida, and T. Mori. 2007. Assist Control Method for Positioning Task Using Master-Slave Manipulators with Flexibility on Slave Arm. In: Proceedings of the IEEE International Conference on Control Applications, pages 232–237.

[105] M. Mulder, M. Mulder, M. M. van Paassen, and D. A. Abbink. 2005. Identification of Driver Car-Following Behaviour. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, volume 3, pages 2905–2910.

[106] T. Murphy, C. Vignes, D. Yuh, and A. Okamura. 2003. Automatic Motion Recognition and Skill Evaluation for Dynamic Tasks. In: Eurohaptics, pages 363–373. Dublin, Ireland. URL `http://cite-seer.ist.psu.edu/murphy03automatic.html`.

[107] Erik Neumann. 2004. Pendulum and Cart Physics Simulation (Cited at May 6 2010). web page. URL `www.myphysicslab.com/pendulum_cart.html`.

[108] F. M. Nieuwenhuizen, P. M. T. Zaal, M. Mulder, and M. M. van Paassen. 2006. A New Multi-Channel Pilot Model Identification Method for Use in Assessment of Simulator Fidelity. In: Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit. Keystone, Colorado.

[109] F.M. Nieuwenhuizen, K.A. Beykirch, M. Mulder, and H.H. Bülthoff. 2007. Identification of Pilot Control Behavior in a Roll-Lateral Helicopter Hover Task. In: Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit. Hilton Head, South Carolina.

[110] I. Nimmo. 1995. Adequately address abnormal operations. Chemical Engineering Progress 91, no. 9, pages 36–45.

[111] F. Nori and R. Frezza. 2004. Linear Optimal Control Problems and Quadratic Cost Function Estimation. In: Proceedings of the 12th Mediterranean Conference on Control and Automation, MED'04.

[112] Donald A. Norman. 1988. The Psychology of Everyday Things. Basic Books, New York, 16 edition.

[113] Sehoon Oh, T. Koyanagi, and Y. Hori. 2009. Force sensor-less assist control design based on two-degree-of-freedom control. In: Proceedings of the International Joint Conference on Instrumentation, Control and Information Technology, ICCAS-SICE, pages 1633–1638.

[114] H. Okuda, H. Takeuchi, S. Inagaki, T. Suzuki, and S. Hayakawa. 2009. Understanding of positioning skill based on feedforward / feedback switched dynamical model. In: Proceedsings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3057–3057.

[115] Hiroyuki Okuda, Hidenori Takeuchi, Shinkichi Inagaki, and Tatsuya Suzuki. 2009. Identification of Human Skill based on Feedforward/Feedback Switched Dynamical Model. In: Proceedings of the IEEE International Joint Conference ICROS-SICE, pages 235–240. Fukuoka, Japan.

[116] F. Osafo-Charles, G. C. Agarwal, W. D. O'Neill, and G. L. Gottlieb. 1980. Application of Time-Series Modeling to Human Operator Dynamics. IEEE Transactions on Systems, Man, and Cybernetics 10, pages 849–860.

[117] H. Ovaskainen, J. Uusitalo, and K. Väätäinen. 2004. Characteristics and Significance of a Harvester Operators' Working Technique in Thinnings. International Journal of Forest Engineering 15, no. 2.

[118] Lauri Palmroth and Aki Putkonen. 2006. Work cycle recognition in human operated machines using hidden Markov models. In: Proceedings of the 8th International Conference on Motion and Vibration Control (MOVIC 2006), pages 459–464. Daejeon, Korea.

[119] Lauri Palmroth, Kalevi Tervo, and Aki Putkonen. 2009. Intelligent coaching of mobile working machine operators. In: Proceedings of the IEEE 13th International Conference on Intelligent Engineering Systems. Barbados.

[120] A. Pekkola. 2009. Development of work cycle analysis system for a container crane (in Finnish). Master's thesis, Helsinki University of Technology, Espoo, Finland.

[121] K. B. Petersen and M. S. Pedersen. 2008. The Matrix Cookbook. URL `http://www2.imm.dtu.dk/pubdb/p.php?3274`. Version 20081110.

[122] A. V. Phatak and G. A. Bekey. 1969. Model of the Adaptive Behavior of the Human Operator in Response to a sudden Change in the Control Situation. IEEE Transactions on Man-Machine Systems 10, pages 72–80.

[123] Janne Pietilä and Olli Haavisto. 2010. Comparison of operator performance in a mineral processing plant. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics SMC2010. Istanbul, Turkey.

[124] Rik Pintelon and Johan Schoukens. 2001. System Identification: A Frequency Domain Approach. IEEE Press, Piscataway, New Jersey, 6 edition.

[125] L. R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, no. 2, pages 257–286.

[126] M. Ralph and M.A. Moussa. 2008. Toward a Natural Language Interface for Transferring Grasping Skills to Robots. IEEE Transactions on Robotics 24, no. 2, pages 468–475.

[127] Jens Rasmussen. 1983. Skills, Rules and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. IEEE Transactions on Systems, Man, and Cybernetics smc-1 3, no. 3.

[128] Petri Rauhala and Heikki N. Koivo. 1994. Modelling and Robust Control of a Telemanipulator Test Bench. Technical report, Tampere University of Technology, Tampere, Finland.

[129] Raimundo Real and Juan M. Vargas. 1996. The Probabilistic Basis of Jaccard's Index of Similarity. Systematic Biology 45, no. 3, pages 380–385.

[130] D. W. Repperger, C. A. Phillips, and M. W. Haas. 1998. A study of adaptive stick controllers in human interface systems. In: Proceedings of the American Control Conference, volume 1, pages 453–455.

[131] J. Rosen, L. Chang, J. D. Brown, B. Hannaford, M. N. Sinanan, and R. Satava. 2003. Minimally invasive surgery task decomposition - etymology of endoscopic suturing. In: Studies in Health Tech. and Inf. - Medic. Meets Virt. Real., volume 94, pages 295–301. IOS Press.

[132] J. Rosen, B. Hannaford, C. G. Richards, and M. N. Sinanan. 2001. Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills. IEEE Transactions on Biomedical Engineering 48, no. 5.

[133] T. Sasaki, A Takeya, H. Igarashi, and S. Suzuki. 2007. Operation skill quantification for mobile vehicle operation. In: Proceedings of SICE 2007 Annual Conference.

[134] David K. Schmidt. 1979. Optimal Flight Control Synthesis Via Pilot Modeling. Journal on Guidance & Control 2, no. 4, pages 306–312.

[135] O. C. Schrempf and U. D. Hanebeck. 2005. A Generic Model for Estimating User-Intentions in Human-Robot Cooperation. In: Proceedings of the 2nd International Conference on Informatics in Control, Automation and Robotics, pages 250–256.

[136] O.C. Schrempf, D. Albrecht, and U.D. Hanebeck. 2007. Tractable probabilistic models for intention recognition based on expert knowledge. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1429–1434.

[137] O.C. Schrempf, U.D. Hanebeck, A.J. Schmid, and H. Worn. 2005. A novel approach to proactive human-robot cooperation. In: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, pages 555–560.

[138] Thomas B. Sheridan. 1992. Telerobotics, Automation, and Human Supervisory Control. The MIT Press, Cambridge, MA.

[139] Thomas B. Sheridan. 2002. Humans and Automation: System Design and Research Issues. John Wiley & Sons, Inc., Publication.

[140] S.M. Shinners. 1974. Modeling of Human Operator Performance Utilizing Time Series Analysis. IEEE Transactions on Systems, Man, and Cybernetics 4, no. 5, pages 446–458.

[141] Dan Simon. 2006. Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley.

[142] D. Steenken, S. Vos, and R. Stahlbock. 2004. Container terminal operation and operations research - a classification and literature review. Springer Berlin Heidelberg.

[143] Indar Sugiarto and Felix Pasila. 2004. Design and implementation of SCADAlarm in a drink production process using wonderware. URL `http://dewey.petra.ac.id/jiunkpe_dg_7406.html`.

[144] S. Suzuki. 2010. Human Adaptive Mechatronics. IEEE Industrial Electronics Magazine 4, no. 2, pages 28–35.

[145] S. Suzuki and F. Harashima. 2006. Assist control and its tuning method for haptic system. In: Proceedings of the 9th IEEE International Workshop on Advanced Motion Control, pages 374–379.

[146] S. Suzuki, K. Kurihara, K. Furuta, F. Harashima, and Y. Pan. 2005. Variable Dynamic Assist Control on Haptic System for Human Adaptive Mechatronics. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference. Seville, Spain.

[147] S. Suzuki, N. Tomomatsu, F. Harashima, and K. Furuta. 2004. Skill evaluation based on state-transition model for human adaptive mechatronics (HAM). In: Proceedings of the 30th IEEE Annual Conference on Industrial Electronics Society, volume 1, pages 641–646.

[148] Satoshi Suzuki. 2009. Visualization of task switching strategy of machine operation. In: Proceedings of the IEEE International Conference on Networking, Sensing and Control, pages 513–518. Okayama, Japan.

[149] K. A. Tahboub. 2006. Intelligent Human-Machine Interaction Based on Dynamic Bayesian Networks Probabilistic Intention Recognition. Journal of Intelligent and Robotic Systems 45, no. 1, pages 31–52.

[150] K.A. Tahboub. 2004. Intention recognition of a human commanding a mobile robot. In: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, volume 2, pages 896–901.

[151] K.A. Tahboub. 2005. Compliant Human-Robot Cooperation Based on Intention Recognition. In: Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation, pages 1417–1422.

[152] W. Takano, K. Yamane, T. Sugihara, K. Yamamoto, and Y. Nakamura. 2006. Primitive communication based on motion recognition and generation with hierarchical mimesis model. In: Proceedings the IEEE International Conference on Robotics and Automation, pages 3602–3609.

[153] T. Takeda, Y. Hirata, and K. Kosuge. 2007. Dance Step Estimation Method Based on HMM for Dance Partner Robot. IEEE Transactions on Industrial Electronics 54, no. 2, pages 699–706.

[154] K. Tervo and H. Koivo. 2010. Towards Human Skill Adaptive Manual Control. International Journal of Advanced Mechatronic Systems 2, no. 1/2, pages 46–48.

[155] Kalevi Tervo. 2006. Evaluation of tree stem feeding process performance using hidden Markov models. Master's thesis, Helsinki University of Technology, Espoo, Finland.

[156] Kalevi Tervo and Vesa Hölttä. 2007. Hidden Markov Models in Step Response Evaluation. In: Proceedings of The 61st Meeting of the Society for Machinery Failure Prevention Technology. Virginia Beach, Virginia, USA.

[157] Kalevi Tervo and Aino Manninen. 2010. Analysis of Model Orders in Human Dynamics Identification using Linear Polynomial and Hammerstein-Wiener Structures. In: IEEE International Conference on Networking, Sensing and Control, pages 614–620. Chicago, IL, USA.

[158] Kalevi Tervo, Lauri Palmroth, Vesa Hölttä, and Aki Putkonen. 2008. Improving Operator Skills with Productivity Model Feedback. In: Proceedings of the 17th World Congress. IFAC.

[159] Kalevi Tervo, Lauri Palmroth, and Heikki N. Koivo. 2010. Skill Evaluation of Human Operators in Partly Automated Mobile Working Machines. IEEE Transactions on Automation Science and Engineering 7, no. 1, pages 133–142.

[160] Kalevi Tervo, Lauri Palmroth, and Aki Putkonen. 2009. A Hierarchical Fuzzy Inference Method for Skill Evaluation of Machine Operators. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 136–141. Singapore.

[161] Sergios Theodoridis and Konstantinos Koutroumbas. 1999. Pattern Recognition. Academic Press.

[162] Emanuel Todorov. 2009. Efficient computation of optimal actions. Proceedings of the National Academy of Sciences of the United States of America 107, no. 28.

[163] I.F.A. Vis and R. de Koster. 2003. Transshipment of containers at a container terminal: An overview. European Journal of Operational Research 147, no. 1, pages 1–16.

[164] A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13, no. 2, pages 260–269.

[165] J. Willems and H. Van De Voorde. 1977. Inverse optimal control problem for linear discrete-time systems. Electronics Letters 13, no. 17.

[166] K. Yacef and L. Alem. 1996. Evaluation of learner's skills in the context of dynamic and complex systems. In: IEEE International Conference on Systems, Man, and Cybernetics, volume 3, pages 2200–2204.

[167] Jie Yang, Yangsheng Xu, and Chiou S. Chen. 1997. Human Action Learning via Hidden Markov Model. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 27, pages 34–44.

[168] M. Yoneda, F. Arai, T. Fukuda, K. Miyata, and T. Naito. 1997. Assistance system for crane operation using multimodal display. In: Proceedings of the IEEE International Conference on Robotics and Automation, volume 1, pages 40–45.

[169] M. Yoneda, F. Arai, T. Fukuda, K. Miyata, and T. Naito. 1997. Operational assistance system for crane using interactive adaptation interface-design of 3D virtual crane simulator for operation training. In: Proceedings of the 6th IEEE International Workshop on Robot and Human Communication, pages 224–229.

[170] M. Yoneda, F. Arai, T. Fukuda, K. Miyata, and T. Naito. 1999. Assistance system for crane operation with haptic display operational assistance to suppress round payload swing. In: Proceedings of the IEEE International Conference on Robotics and Automation, volume 4, pages 2924–2929.

[171] M. Yoneda, F. Arai, T. Fukuda, K. Miyata, and T. Naito. 1999. VR training system with adaptive operational assistance considering straight-line transfer operation. In: Proceedings of the 8th IEEE International Workshop on Robot and Human Interaction, pages 142–147.

[172] So-Jeong Youn and Kyung-Whan Oh. 2007. Intention recognition using a graph representation. International Journal of Applied Science, Engineering and Techcnology 4, pages 13–18.

[173] W. Yu, R. Dubey, and N. Pernalete. 2004. Robotic therapy for persons with disabilities using hidden Markov model based skill learning. In: IEEE International Conference on Robotics and Automation, volume 2, pages 2074–2079.

[174] Wentao Yu, R. Alqasemi, Rajiv Dubey, and Norali Pernalete. 2005. Telemanipulation Assistance Based on Motion Intention Recognition. In: Proceedings of the IEEE International Conference on Robotics and Automation, pages 1121–1126.

[175] L. A. Zadeh. 1965. Fuzzy sets. Information and Control 8, no. 3.

Appendix A Code for work cycle simulation example

A.1 Data generation function

```
function [X,I,E,Et,Ei] = diss_generateData
% This function generates data for an imaginary work cycle recognition
% example. The work cycle recognition is based on three continuous signals.
% From the signals, four events are extracted for the work cycle model. The
% events are based in the level combinations of the three signals.
%
% Inputs: none
% Outputs:
%           X - The continuous signals.
%           I - The true state indices.
%           E - The event sequence.
%           Et - The event time instants.
%           Ei - The state flags corresponding the events.
%
% Author: Kalevi Tervo
% Last modification: 4.4.2010, Kalevi Tervo

% Generate example signals
inc = 2.5;
N = 40;
x1 = [randn(N,1) randn(N,1) randn(N,1)];
i1 = ones(length(x1),1);
x2 = [randn(N,1)+inc randn(N,1) randn(N,1)];
i2 = 2*ones(length(x2),1);
x3 = [randn(N,1)+inc randn(N,1)+inc randn(N,1)];
i3 = 3*ones(length(x3),1);
x4 = [randn(N,1)+inc randn(N,1)+inc randn(N,1)+inc];
i4 = 4*ones(length(x4),1);

X = [x1; x2; x3; x4];

high_thresh = 2.1;
% Define 4 events
E1 = 1; % all low
E2 = 2; % signal 1 high
E3 = 3; % signals 1 and 2 high
E4 = 4; % all high

I = [i1; i2; i3; i4];
yLims = [min(min(X)) max(max(X))];

E = [];
Et = [];
Ei = [];
event_now = 1;
for i=1:length(I)
    if X(i,1)<high_thresh && X(i,2)<high_thresh && X(i,3)<high_thresh
```

```
        event_now = E1;
    elseif X(i,1)>high_thresh && X(i,2)<high_thresh && X(i,3)<high_thresh
        event_now = E2;
    elseif X(i,1)>high_thresh && X(i,2)>high_thresh && X(i,3)<high_thresh
        event_now = E3;
    elseif X(i,1)>high_thresh && X(i,2)>high_thresh && X(i,3)>high_thresh
        event_now = E4;
    end
    E = [E event_now];
    Et = [Et i];
    Ei = [Ei I(i)];
end
```

A.2 Work cycle model training and validation

```
% This script can be used to set up and test an imaginary example for
% work cycle recognition using hidden Markov models. Both, supervised and
% unsupervised learning methods are implemented for the modeling. In
% addition, the confusion matrices are computed to test the reliability of
% the modeling methods.
%
% Author: Kalevi Tervo
% Last modification: 4.4.2010, Kalevi Tervo

%% Get the training data for work cycle modeling
% Generate some data for training (say 10 sequences)
for k=1:10
    [tempX{k},tempI{k},tempE{k},tempEt{k},tempEi{k}] = diss_generateData;
end

%% Estimation problem: estimate the parameters of the work cycle model
%% Supervised frequency-based training
Af = zeros(4);
Bf = zeros(4);
for k=1:length(tempX)
    for i=2:length(tempX{k})
        Af(tempI{k}(i-1),tempI{k}(i)) = Af(tempI{k}(i-1),tempI{k}(i)) + 1;
        Bf(tempI{k}(i),tempE{k}(i)) = Bf(tempI{k}(i),tempE{k}(i)) + 1;
    end
end
% Make the matrices stochastic
A = Af*diag(1./sum(Af,2));
B = Bf*diag(1./sum(Bf,2));
% For numerical stability
B(B==0) = 1e-8;
A(A==0) = 1e-12;

%% Unsupervised Viterbi re-estimation training
% Initial guesses (state transition matrix needs to be highly diagonal)
Ag = [0.99 0.01 0 0;
    0 0.99 0.01 0;
```

```
    0 0 0.99 0.01;
    0 0 0 1];
Bg = [0.8 0.2 0 0;
    0.2 0.7 0.1 0;
    0.1 0.2 0.6 0.1
    0.1 0.1 0.2 0.6];

% Perform training by using the Viterbi re-estimation procedure
[Av,Bv] = hmmtrain(tempE,Ag,Bg,'algorithm','viterbi');
% For numerical stability
Bv(Bv==0) = 1e-8;
Av(Av==0) = 1e-12;

%% Generate another set of data and test the trained model and compute the
%% confusion matrices
CM = zeros(4);
CMv = zeros(4);
% Let's generate about 10 sequences of data for validation
for k=1:10
    % Get the data
    [X2{k},I2{k},E2{k},Et2{k},Ei2{k}] = diss_generateData;
    % Decoding: most likely state sequences
    states{k} = hmmviterbi(E2{k},A,B)'; % Supervised model
    statesv{k} = hmmviterbi(E2{k},Av,Bv)'; % Unsupervised model
    %% Compute the confusion matrices
    % Supervised model
    for i=1:4
        for j=1:4
            CM(j,i) = CM(j,i) + sum(I2{k}==j & states{k}==i);
        end
    end
    % Unsupervised model
    for i=1:4
        for j=1:4
            CMv(j,i) = CMv(j,i) + sum(I2{k}==j & statesv{k}==i);
        end
    end
end

% Normalize the row sums
CMp = CM*diag(1./sum(CM,2))
CMpv = CMv*diag(1./sum(CMv,2))
```

Appendix B MOCM Details

B.1 Modified optimal control model of human operator

In the MOCM framework the main assumption is that a motivated and well-trained human operator controls a dynamical system as an optimal controller [80, 19]. This is because during training the operator learns the dynamics of the controlled process, and thus can predict the behavior of the system for a given input. The MOCM was developed by Davidson and Smith based on the OCM. A theoretical framework for MOCM is given in [32] from which all related formulae in Appendix B.1 and B.2 are cited. In the MOCM the time delay in the human operation is replaced with Padé approximation. Therefore, the resulting transfer function consists only zeros and poles and is thus easier to analyze than OCM.

The structure of the MOCM is shown in Figure B.1. The human operator perceives the outputs $\mathbf{y}$ ($\mathbf{y} \in \mathbb{R}^{N_y}$) of the system. Due to the uncertainty inherent in the human senses the measurement is corrupted by noise process $\mathbf{v}_y$ ($\mathbf{v}_y \in \mathbb{R}^{N_y}$) with intensity $\mathbf{V}_y$ ($\mathbf{V}_y \in \mathbb{R}^{N_y \times N_y}$). The operator estimates the current states (state estimate $\hat{\mathbf{x}} \in \mathbb{R}^{N_s}$) of the system based on his/her internal model as well as observations and past control actions $u_c \in \mathbb{R}$. The estimator is assumed to be the Kalman filter. The operator dynamics has limitations which are modeled by the first order neuromotor lag $\frac{1}{\tau_n s+1}$. The operator's control signal u_p is also corrupted by noise $v_u \in \mathbb{R}$. The final stage is the time delay which, in MOCM case, is the Padé approximation of the true time delay, which provides the output of the model $u_d \in \mathbb{R}$. The realized control signal fed to the plant is $\delta \in \mathbb{R}$. In general, one can use also high-order approximations of the time delay, but the second-order approximation already provides very accurate results.

The controlled plant in Figure B.1 is denoted with the following linear state-space representation (with $\mathbf{x}(0) = \mathbf{x}_0$))

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\delta + \mathbf{E}\mathbf{w} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\delta \end{cases}, \tag{B.1}$$

where $\mathbf{x} \in \mathbb{R}^{N_s}$ are the states of the cystem, $\mathbf{y} \in \mathbb{R}^{N_y}$ are the measurements, and $\mathbf{w}(t)$ ($\mathbf{w}(t) \in \mathbb{R}^{N_s}$) is zero-mean Gaussian white noise process with intensity $\mathbf{W}$ ($\mathbf{W} \in \mathbb{R}^{N_s \times N_s}$). Matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{E}$, $\mathbf{C}$, and $\mathbf{D}$ have appropriate dimensions.

The operator's observation $\mathbf{y}_{\text{obs}} \in \mathbb{R}^{N_y}$ is corrupted by the observation noise $\mathbf{v}_y \in \mathbb{R}^{N_y}$, which is zero-mean Gaussian with the intensity $\mathbf{V}_y$. The operator's observation $\mathbf{y}_{\text{obs}}$ can be written as

$$\mathbf{y}_{\text{obs}} = \mathbf{C}\mathbf{x} + \mathbf{D}\delta + \mathbf{v}_y \tag{B.2}$$

The operator's reaction time delay inherent in the human operation is modeled with Padé approximation for control and estimation synthesis. The second order Padé approximation of $e^{-\tau_p s}$ is [72]

$$\frac{U_d(s)}{U_p(s)} = \frac{1 - \frac{\tau_p s}{2} + \frac{\tau_p^2 s^2}{12}}{1 + \frac{\tau_p s}{2} + \frac{\tau_p^2 s^2}{12}}, \tag{B.3}$$

where $U_d(s)$ and $U_p(s)$ are Laplace transformations of the delayed operator control signal u_d and operator control signal u_p, respectively. The second order Padé approximation can be

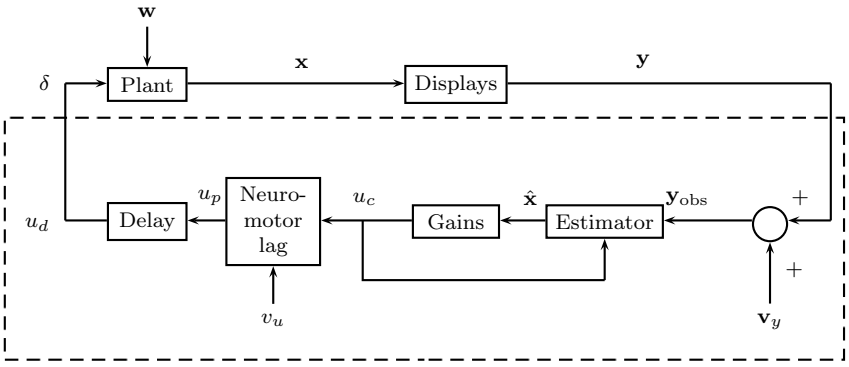


Figure B.1: A block diagram of the modified optimal control model (adapted from [32]). All blocks and signals inside the box marked with dotted line represent the human operator.

described with the following state-space representation.

$$\begin{cases} \begin{bmatrix} \dot{x}_{d1} \\ \dot{x}_{d2} \end{bmatrix} = \begin{bmatrix} -\frac{6}{\tau_p} & -\frac{1}{\tau_p} \\ \frac{12}{\tau_p} & 0 \end{bmatrix} \begin{bmatrix} x_{d1} \\ x_{d2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_p \\ u_d = \begin{bmatrix} -\frac{12}{\tau_p} & 0 \end{bmatrix} \begin{bmatrix} x_{d1} \\ x_{d2} \end{bmatrix} + u_p \end{cases} \tag{B.4}$$

or (observe from Figure B.1 that $u_d = \delta$) rewritten in the matrix form

$$\begin{cases} \dot{\mathbf{x}}_d = \mathbf{A}_d \mathbf{x}_d + \mathbf{B}_d u_p \\ \delta = u_d = \mathbf{C}_d \mathbf{x}_d + u_p \end{cases}, \tag{B.5}$$

where $\mathbf{x}_d \in \mathbb{R}^{N_d}$ (now $N_d = 2$) is the state vector of the delay approximation system and $\mathbf{A}_d = \begin{bmatrix} -\frac{6}{\tau_p} & -\frac{1}{\tau_p} \\ \frac{12}{\tau_p} & 0 \end{bmatrix}$, $\mathbf{B}_d = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and $\mathbf{C}_d = \begin{bmatrix} -\frac{12}{\tau_p} & 0 \end{bmatrix}$. The initial values of (B.5) are $\mathbf{x}_d(0) = \mathbf{x}_{d,0}$.

When the plant (B.1) and the delay model (B.5) is augmented into the same system, the S-plant is obtained

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{C}_d \\ 0 & \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_d \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_d \end{bmatrix} u_p + \begin{bmatrix} \mathbf{E} \\ 0 \end{bmatrix} \mathbf{w} \\ \mathbf{y} = \begin{bmatrix} \mathbf{C} & \mathbf{D}\mathbf{C}_d \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_d \end{bmatrix} + \mathbf{D} u_p \end{cases}$$

or

$$\begin{cases} \dot{\mathbf{x}}_s = \mathbf{A}_s \mathbf{x}_s + \mathbf{B}_s u_p + \mathbf{E}_s \mathbf{w} \\ \mathbf{y} = \mathbf{C}_s \mathbf{x}_s + \mathbf{D}_s u_p \end{cases}, \tag{B.6}$$

with obvious matrix notations, and $\mathbf{x}_s^T(0) = \begin{bmatrix} \mathbf{x}_0^T & \mathbf{x}_{d,0}^T \end{bmatrix}$. The operator's observation vector based on the S-plant can be written as follows

$$\mathbf{y}_{\text{obs}} = \mathbf{C}_s \mathbf{x}_s + \mathbf{D}_s u_p + \mathbf{v}_y$$

MOCM assumes that the operator minimizes the performance index

$$J_p = \lim_{t\to\infty} \frac{1}{t} \int_0^t \left(\mathbf{y}^T \mathbf{Q}_y \mathbf{y} + u_p^2 r + \dot{u}_p^2 f \right) dt \tag{B.7}$$

while executing a given task with $\mathbf{Q}_y \geq 0$ ($\mathbf{Q}_y \in \mathbb{R}^{N_y \times N_y}$), $r \geq 0$ ($r \in \mathbb{R}$), and $f > 0$ ($f \in \mathbb{R}$). Thus, when the number of trials increases, the average response of the human operator approaches the solution of the optimal control problem.

When the S-plant (B.6) is written in the control rate formulation, the 0-plant is obtained as follows

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_d \\ \dot{u}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{C}_d & \mathbf{B} \\ 0 & \mathbf{A}_d & \mathbf{B}_d \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_d \\ u_p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{u}_p + \begin{bmatrix} \mathbf{E} \\ 0 \\ 0 \end{bmatrix} \mathbf{w} \\ \mathbf{y}_{\text{obs}} = \begin{bmatrix} \mathbf{C} & \mathbf{D}\mathbf{C}_d & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_d \\ u_p \end{bmatrix} + \mathbf{v}_y \end{cases}$$

or

$$\begin{cases} \dot{\boldsymbol{\chi}} = \mathbf{A}_0 \boldsymbol{\chi} + \mathbf{B}_0 \dot{u}_p + \mathbf{E}_0 \mathbf{w} \\ \mathbf{y}_{\text{obs}} = \mathbf{C}_0 \boldsymbol{\chi} + \mathbf{v}_y \end{cases}, \tag{B.8}$$

where $\boldsymbol{\chi} = \begin{bmatrix} \mathbf{x}^T & \mathbf{x}_d^T & u_p \end{bmatrix}^T$, $\boldsymbol{\chi}^T(0) = \begin{bmatrix} \mathbf{x}_0^T & \mathbf{x}_{d,0}^T & u_{p,0} \end{bmatrix}$ or compactly $\boldsymbol{\chi}(0) = \boldsymbol{\chi}_0$, and matrix notations are obvious. The initial value of the operator control signal is $u_p(0) = u_{p,0}$. The 0-plant represents the operator's internal model with which the optimal control is synthesized. It does not include the effect of neuromotor lag induced by the process of realizing the human movement, because the control signal is produced by the human brain controller and the desired movement is realized by the human neuro-muscular system. In MOCM only the state estimator accounts for the finite neuromotor lag.

When the performance index (B.7) is minimized with the 0-plant by using the well known LQG technique, the optimal control is ($N_a = N_s + N_d$)

$$\dot{u}_p^* = -\mathbf{g}_p \hat{\boldsymbol{\chi}} = -\begin{bmatrix} g_1 \ldots g_{N_a} \; g_{N_a+1} \end{bmatrix} \hat{\boldsymbol{\chi}} = -f^{-1} \mathbf{B}_0^T \mathbf{K} \hat{\boldsymbol{\chi}}, \tag{B.9}$$

where $\hat{\boldsymbol{\chi}}$ is the optimal state estimate, and $\mathbf{K}$ is the unique positive definite solution of the Riccati equation

$$0 = \mathbf{A}_0^T \mathbf{K} + \mathbf{K}\mathbf{A}_0 + \mathbf{Q}_0 - \mathbf{K}\mathbf{B}_0 f^{-1} \mathbf{B}_0^T \mathbf{K}, \tag{B.10}$$

where

$$\mathbf{Q}_0 = \begin{bmatrix} \mathbf{C}_s^T \mathbf{Q}_y \mathbf{C}_s & \mathbf{C}_s^T \mathbf{Q}_y \mathbf{D}_s \\ \mathbf{D}_s^T \mathbf{Q}_y \mathbf{C}_s & \mathbf{D}_s^T \mathbf{Q}_y \mathbf{D}_s + r \end{bmatrix} \tag{B.11}$$

Thus, the optimal control law (B.9) can be written as ($\hat{\mathbf{x}}_s^T = \begin{bmatrix} \hat{\mathbf{x}}^T & \mathbf{x}_d^T \end{bmatrix}$)

$$\dot{u}_p^* = -\begin{bmatrix} g_1 & \ldots & g_{N_a} \end{bmatrix} \hat{\mathbf{x}}_s - g_{N_a+1} u_p^* \tag{B.12}$$

Moreover, if $\tau_n = \frac{1}{g_{N_a+1}}$ and $\mathbf{l}_p = \tau_n \begin{bmatrix} g_1 \ldots g_{N_a} \end{bmatrix}$, (B.12) can be written as

$$\tau_n \dot{u}_p^* + u_p^* = u_c \tag{B.13}$$

where u_c is defined as

$$u_c = -\mathbf{l}_p \hat{\mathbf{x}}_s \tag{B.14}$$

The weighting parameter f is chosen so that the condition $\tau_n = \frac{1}{g_{N_a+1}}$ is fulfilled. Basically the problem is easiest to solve by using some numerical optimization algorithm. The LMI-based solution is described in Appendix B.4.

The operator's control u_p is corrupted by zero-mean Gaussian white noise $v_u(t)$ with intensity $\mathbf{V}_u$. Since the MOCM assumes that the controller gains are not affected by the control noise, the solution given by the MOCM is only suboptimal. Nevertheless, $\dot{u}_p$ can now be written from (B.13) as

$$\dot{u}_p = \frac{-1}{\tau_n} u_p + \frac{1}{\tau_n} u_c + \frac{1}{\tau_n} v_u \tag{B.15}$$

Now the suboptimal control law can be inserted into the 0-plant (B.8) and the 1-plant is obtained as follows.

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_d \\ \dot{u}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{C}_d & \mathbf{B} \\ 0 & \mathbf{A}_d & \mathbf{B}_d \\ 0 & 0 & -\frac{1}{\tau_n} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_d \\ u_p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau_n} \end{bmatrix} u_c + \begin{bmatrix} \mathbf{E} & 0 \\ 0 & 0 \\ 0 & \frac{1}{\tau_n} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ v_u \end{bmatrix} \\ \mathbf{y}_{\text{obs}} = \begin{bmatrix} \mathbf{C} & \mathbf{D}\mathbf{C}_d & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_d \\ u_p \end{bmatrix} + \mathbf{v}_y \end{cases}$$

or

$$\begin{cases} \dot{\boldsymbol{\chi}} = \mathbf{A}_1 \boldsymbol{\chi} + \mathbf{B}_1 u_c + \mathbf{E}_1 \mathbf{w}_1 \\ \mathbf{y}_{\text{obs}} = \mathbf{C}_1 \boldsymbol{\chi} + \mathbf{v}_y \end{cases} \tag{B.16}$$

with obvious matrix notations, $\mathbf{w}_1^T = \begin{bmatrix} \mathbf{w}^T & v_u \end{bmatrix}$, and $\boldsymbol{\chi}^T(0) = \begin{bmatrix} \mathbf{x}_0^T & \mathbf{x}_{d,0}^T & u_{p,0} \end{bmatrix}$, or compactly $\boldsymbol{\chi}(0) = \boldsymbol{\chi}_0$. The 1-plant is used to synthesize the Kalman filter by which the operator estimates the states of the system.

Using (B.16) the current estimate of the state $\hat{\boldsymbol{\chi}}^T = \begin{bmatrix} \hat{\mathbf{x}}^T & \hat{\mathbf{x}}_d^T & \hat{u}_p \end{bmatrix}$ is given by a Kalman filter

$$\dot{\hat{\boldsymbol{\chi}}} = \mathbf{A}_1 \hat{\boldsymbol{\chi}} + \mathbf{B}_1 u_c + \mathbf{F} \left(\mathbf{y}_{\text{obs}} - \hat{\mathbf{y}} \right) \tag{B.17a}$$

$$\dot{\hat{\boldsymbol{\chi}}} = \left(\mathbf{A}_1 - \mathbf{F}\mathbf{C}_1 \right) \hat{\boldsymbol{\chi}} + \mathbf{F}\mathbf{C}_1 \boldsymbol{\chi} + \mathbf{B}_1 u_c + \mathbf{F}\mathbf{v}_y \tag{B.17b}$$

where

$$\mathbf{F} = \boldsymbol{\Sigma}_1 \mathbf{C}_1^T \mathbf{V}_y^{-1} \tag{B.18}$$

and $\boldsymbol{\Sigma}_1$ is the positive definite solution of the Riccati equation

$$0 = \mathbf{A}_1 \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_1 \mathbf{A}_1^T + \mathbf{W}_1 - \boldsymbol{\Sigma}_1 \mathbf{C}_1^T \mathbf{V}_y^{-1} \mathbf{C}_1 \boldsymbol{\Sigma}_1, \tag{B.19}$$

with $\mathbf{W}_1 = \text{diag}(\mathbf{W}, \mathbf{V}_u)$, $\mathbf{W} \geq 0$, $\mathbf{V}_u \geq 0$, and $\mathbf{V}_y > 0$. The initial values of the Kalman filter are defined as $\hat{\boldsymbol{\chi}}^T(0) = \begin{bmatrix} \hat{\mathbf{x}}(0)^T & \hat{\mathbf{x}}_d(0)^T & \hat{u}_p(0) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_0^T & \hat{\mathbf{x}}_{d,0}^T & \hat{u}_{p,0} \end{bmatrix}$, where $\hat{\mathbf{x}}_0$, $\hat{\mathbf{x}}_{d,0}$ and $\hat{u}_{p,0}$ are the initial values of the plant state estimates, delay state estimates and the operator control, respectively. More compactly the initial values can be expressed using $\hat{\boldsymbol{\chi}}(0) = \hat{\boldsymbol{\chi}}_0$.

Let $u_c = \mathbf{l}_p \hat{\mathbf{x}}_s = \mathbf{l}_1 \hat{\boldsymbol{\chi}}$, where $\mathbf{l}_1 = \begin{bmatrix} \mathbf{l}_p & 0 \end{bmatrix}$. Based on the MOCM block diagram shown in Figure B.1 and (B.17b) the operator's dynamics can be described with the following state-

space representation

$$\begin{cases} \begin{bmatrix} \dot{\hat{\boldsymbol{\chi}}} \\ \dot{u}_p \\ \dot{\mathbf{x}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 - \mathbf{F}\mathbf{C}_1 - \mathbf{B}_1\mathbf{l}_1 & 0 & 0 \\ \frac{\mathbf{l}_1}{\tau_n} & -\frac{1}{\tau_n} & 0 \\ 0 & \mathbf{B}_d & \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\chi}} \\ u_p \\ \mathbf{x}_d \end{bmatrix} + \begin{bmatrix} \mathbf{F} \\ 0 \\ 0 \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{F} & 0 \\ 0 & \frac{1}{\tau_n} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_y \\ v_u \end{bmatrix} \\ \delta = \begin{bmatrix} 0 \, 1 \, \mathbf{C}_d \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\chi}} \\ u_p \\ \mathbf{x}_d \end{bmatrix} \end{cases}$$

or

$$\begin{cases} \dot{\mathbf{x}}_p = \mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\mathbf{y} + \mathbf{E}_p\mathbf{v}_p \\ \delta = \mathbf{C}_p\mathbf{x}_p \end{cases} \tag{B.20}$$

with obvious matrix notations and the initial values $\mathbf{x}_p^T(0) = \begin{bmatrix} \hat{\boldsymbol{\chi}}_0^T & u_{p,0} & \mathbf{x}_{d,0}^T \end{bmatrix}$.

The closed loop noise driven man-machine system including the plant and the MOCM is

$$\begin{cases} \begin{bmatrix} \dot{\boldsymbol{\chi}} \\ \dot{\hat{\boldsymbol{\chi}}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & -\mathbf{B}_1\mathbf{l}_1 \\ \mathbf{F}\mathbf{C}_1 & \mathbf{A}_1 - \mathbf{B}_1\mathbf{l}_1 - \mathbf{F}\mathbf{C}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\chi} \\ \hat{\boldsymbol{\chi}} \end{bmatrix} + \begin{bmatrix} \mathbf{E}_1 & 0 \\ 0 & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{v}_y \end{bmatrix} \\ \begin{bmatrix} \mathbf{y}_{\text{obs}} \\ \delta \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 & 0 \\ \mathbf{C}_\delta & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\chi} \\ \hat{\boldsymbol{\chi}} \end{bmatrix} \end{cases}, \tag{B.21}$$

where $\mathbf{C}_\delta = \begin{bmatrix} 0 & \mathbf{C}_d & 1 \end{bmatrix}$, and the initial values $\begin{bmatrix} \boldsymbol{\chi}^T(0) & \hat{\boldsymbol{\chi}}^T(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\chi}_0^T & \hat{\boldsymbol{\chi}}_0^T \end{bmatrix}$. Given the neuromotor time constant τ_n, state weighting matrix $\mathbf{Q}_0$, and the noise intensities $\mathbf{W}_1$ and $\mathbf{V}_y$, the numerical values for state feedback gain $\mathbf{g}_p$ and the Kalman filter gain $\mathbf{F}$ can be obtained by using the LMI approach described in B.4.

B.2 Evaluation of the performance index for noise driven system

The theoretical value for the performance index of a noise driven closed loop human-machine system with MOCM was derived in [34], from where the following is cited. The (0-plant) performance index (B.7) is equal to the 1-plant performance index

$$J_p = \lim_{t\to\infty} \frac{1}{t} \int_0^t \left(\boldsymbol{\chi}^T\mathbf{R}_1\boldsymbol{\chi} + 2\boldsymbol{\chi}^T\mathbf{R}_{12}u_c + u_c^T\mathbf{R}_2u_c \right) dt, \tag{B.22}$$

where

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{C}_s^T\mathbf{Q}_y\mathbf{C}_s & \mathbf{C}_s^T\mathbf{Q}_y\mathbf{D}_s \\ \mathbf{D}_s^T\mathbf{Q}_y\mathbf{C}_s & \mathbf{D}_s^T\mathbf{Q}_y\mathbf{D}_s + r + \frac{f}{\tau_n^2} \end{bmatrix} \quad \mathbf{R}_{12} = \begin{bmatrix} 0 & -\frac{f}{\tau_n^2} \end{bmatrix} \quad \mathbf{R}_2 = \frac{f}{\tau_n^2} \tag{B.23}$$

and matrices are as $\mathbf{R}_1 \geq 0$ and $\mathbf{R}_2 > 0$. Let the integrand of the performance index to be defined as

$$J_Q = \boldsymbol{\chi}^T\mathbf{R}_1\boldsymbol{\chi} + 2\boldsymbol{\chi}^T\mathbf{R}_{12}u_c + u_c^T\mathbf{R}_2u_c. \tag{B.24}$$

By using $u_c = -\mathbf{l}_1\hat{\boldsymbol{\chi}}$, one obtains

$$J_Q = \boldsymbol{\chi}^T\mathbf{R}_1\boldsymbol{\chi} - 2\boldsymbol{\chi}^T\mathbf{R}_{12}\mathbf{l}_1\hat{\boldsymbol{\chi}} + \hat{\boldsymbol{\chi}}^T\mathbf{l}_1^T\mathbf{R}_2\mathbf{l}_1\hat{\boldsymbol{\chi}}, \tag{B.25}$$

which is equivalent with

$$J_Q = \mathrm{tr}\left(J_Q\right) = \mathrm{tr}\left(\begin{bmatrix}\boldsymbol{\chi}^T & \hat{\boldsymbol{\chi}}^T\end{bmatrix}\begin{bmatrix}\mathbf{R}_1 & -\mathbf{R}_{12}\mathbf{l}_1 \\ -\mathbf{l}_1^T\mathbf{R}_{12}^T & \mathbf{l}_1^T\mathbf{R}_2\mathbf{l}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{\chi} \\ \hat{\boldsymbol{\chi}}\end{bmatrix}\right). \tag{B.26}$$

By using the trace identity $\mathrm{tr}\left(\mathbf{AB}\right) = \mathrm{tr}\left(\mathbf{BA}\right)$ [121] one obtains

$$\mathrm{tr}\left(J_Q\right) = \mathrm{tr}\left(\begin{bmatrix}\boldsymbol{\chi} \\ \hat{\boldsymbol{\chi}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\chi}^T & \hat{\boldsymbol{\chi}}^T\end{bmatrix}\begin{bmatrix}\mathbf{R}_1 & -\mathbf{R}_{12}\mathbf{l}_1 \\ -\mathbf{l}_1^T\mathbf{R}_{12}^T & \mathbf{l}_1^T\mathbf{R}_2\mathbf{l}_1\end{bmatrix}\right), \tag{B.27}$$

that is

$$\mathrm{tr}\left(J_Q\right) = \mathrm{tr}\left(\mathbf{X}\tilde{\mathbf{R}}\right) \tag{B.28}$$

where

$$\tilde{\mathbf{R}} = \begin{bmatrix}\mathbf{R}_1 & -\mathbf{R}_{12}\mathbf{l}_1 \\ -\mathbf{l}_1^T\mathbf{R}_{12}^T & \mathbf{l}_1^T\mathbf{R}_2\mathbf{l}_1\end{bmatrix} \quad \text{and} \quad \mathbf{X} = \begin{bmatrix}\boldsymbol{\chi}\boldsymbol{\chi}^T & \boldsymbol{\chi}\hat{\boldsymbol{\chi}}^T \\ \hat{\boldsymbol{\chi}}\boldsymbol{\chi}^T & \hat{\boldsymbol{\chi}}\hat{\boldsymbol{\chi}}^T\end{bmatrix}. \tag{B.29}$$

Because

$$J_p = \lim_{t\to\infty}\frac{1}{t}\int_0^t J_Q d\tau = \lim_{t\to\infty}\frac{1}{t}\int_0^t \mathrm{tr}\left(\mathbf{X}\tilde{\mathbf{R}}\right)d\tau = \mathrm{tr}\left(\lim_{t\to\infty}\frac{1}{t}\int_0^t \mathbf{X}d\tau\tilde{\mathbf{R}}\right), \tag{B.30}$$

the performance index can be evaluated using

$$J_p = \mathrm{tr}\left(\tilde{\mathbf{X}}\tilde{\mathbf{R}}\right), \tag{B.31}$$

where $\tilde{\mathbf{X}}$ is the solution to the Lyapunov equation of the noise driven closed loop system (B.21), that is,

$$\tilde{\mathbf{A}}\tilde{\mathbf{X}} + \tilde{\mathbf{X}}\tilde{\mathbf{A}}^T + \tilde{\mathbf{V}} = 0. \tag{B.32}$$

The matrices for the Lyapunov equation are

$$\tilde{\mathbf{A}} = \begin{bmatrix}\mathbf{A}_1 & -\mathbf{B}_1\mathbf{l}_1 \\ \mathbf{F}\mathbf{C}_1 & \mathbf{A}_1 - \mathbf{B}_1\mathbf{l}_1 - \mathbf{F}\mathbf{C}_1\end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{V}} = \begin{bmatrix}\mathbf{E}_1\mathbf{W}_1\mathbf{E}_1^T & 0 \\ 0 & \mathbf{F}\mathbf{V}_y\mathbf{F}^T\end{bmatrix}. \tag{B.33}$$

B.3 MOCM adaptation example

The MOCM parameters for the adaptation example in Section 6.1 are shown in Table B.2. Using the parameters, the MOCM feedback and gain matrices can be solved using the standard MOCM procedure described in Appendix B.1. The matrices $\mathbf{A}$ and $\mathbf{B}$ for the trolley crane model of different cases (baseline, double mass and longer rope) are shown in Table B.3. The parameters are obtained using the values shown in Table B.1. The other matrices of the plant are

$$\mathbf{C} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} \tag{B.34}$$

$$\mathbf{D} = 0. \tag{B.35}$$

Parameter	Baseline case	Double mass case	Longer rope case
m	2.00 kg	4.00 kg	2.00 kg
M	1.00 kg	1.00 kg	1.00 kg
L	1.00 m	1.00 m	1.40 m
b	1.00 $\frac{\text{Ns}}{\text{m}}$	1.00 $\frac{\text{Ns}}{\text{m}}$	1.00 $\frac{\text{Ns}}{\text{m}}$

Table B.1: Physical parameters of the trolley crane model in the MOCM adaptation example in Section 6.1.

Parameter	Value
τ_n	0.15
τ_p	0.40
r	0
V_u	0.50
$\mathbf{V}_y$	0.5
$\mathbf{Q}_y$	1
$\mathbf{E}$	$\mathbf{I}_{4\times4}$
$\mathbf{W}$	$\mathbf{I}_{4\times4}$

Table B.2: The MOCM parameters for the adaptation example in Section 6.1

Case	A	B
Baseline	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 19.62 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & -29.43 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$
Double mass	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 39.24 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & -49.05 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$
Longer rope	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 19.62 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.7143 & -21.0214 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \\ -0.7143 \end{bmatrix}$

Table B.3: The parameters of the trolley crane in the MOCM adaption example in Section 6.1

B.4 LMI solution for control and estimator gains

In MOCM the essential problem is to solve the optimal control gain vector $\mathbf{g}_p$ and the Kalman filter gain matrix $\mathbf{F}$ given the 0- and 1-plants, weighting matrices $\mathbf{Q}_0$ and $\mathbf{W}_1$, and the neuro-

motor time constant τ_n. The optimal control gain vector can be solved using (B.9) and (B.10), while minimizing

$$\left(\mathbf{B}_0^T\mathbf{K}\mathbf{D}_1 - \frac{f}{\tau_n}\right)^2, \tag{B.36}$$

where $\mathbf{D}_1 = \begin{bmatrix}0 & 0 & \dots & 1\end{bmatrix}^T$ ($\mathbf{D}_1 \in \mathbb{R}^{N_s+1}$). Minimization of (B.36) guarantees that $g_{N_s+1} = \frac{1}{\tau_n}$. Considering the Riccati equations (B.10) and (B.19), and minimization of (B.36), by using Schur complements [121] the following LMIs can be defined

$$\mathbf{L}_1 = \begin{bmatrix}\mathbf{A}_0^T\mathbf{K} + \mathbf{K}\mathbf{A}_0 + \mathbf{Q}_0 & \mathbf{K}\mathbf{B}_0 \\ \mathbf{B}_0^T\mathbf{K} & f\end{bmatrix} \geq 0 \tag{B.37}$$

$$\mathbf{L}_2 = \begin{bmatrix}\lambda & \left(\mathbf{B}_0\mathbf{K}\mathbf{D}_1 - \frac{f}{\tau_n}\right)^T \\ \mathbf{B}_0\mathbf{K}\mathbf{D}_1 - \frac{f}{\tau_n} & 1\end{bmatrix} \geq 0 \tag{B.38}$$

$$\mathbf{L}_3 = \begin{bmatrix}\mathbf{A}_1\mathbf{\Sigma}_1 + \mathbf{\Sigma}_1\mathbf{A}_1^T + \mathbf{W}_1 & \mathbf{\Sigma}_1\mathbf{C}_1^T \\ \mathbf{C}_1\mathbf{\Sigma}_1 & \mathbf{V}_y\end{bmatrix} \geq 0, \tag{B.39}$$

where $\mathbf{L}_1 \geq 0$ corresponds to the Riccati inequality derived from (B.10), $\mathbf{L}_2 \geq 0$ corresponds to minimization of (B.36), and $\mathbf{L}_3 \geq 0$ corresponds to the Riccati inequality derived from (B.19).

Now, given $\mathbf{Q}_0$, $\mathbf{W}_1$ and τ_n (and 0- and 1-plants) the solution for $\mathbf{g}_p$ and $\mathbf{F}$ can be obtained by (B.9) and (B.18), where $\mathbf{K}$ and $\mathbf{\Sigma}_1$ are the positive definite solutions for the LMIP

$$\begin{aligned}&\min_{f,\mathbf{K},\mathbf{\Sigma}_1,\lambda} \; w_\lambda\lambda - w_K\mathrm{tr}\left(\mathbf{K}\right) - w_\Sigma\mathrm{tr}\left(\mathbf{\Sigma}_1\right) \\ &\text{s.t. } \mathbf{L}_1 \geq 0, \mathbf{L}_2 \geq 0, \mathbf{L}_3 \geq 0, \mathbf{K} \geq 0, \mathbf{\Sigma}_1 \geq 0, f > 0.,\end{aligned} \tag{B.40}$$

where w_λ, w_K and w_Σ are positive weights for different objectives. Thus, the LMIP is solved when maximal positive definite $\mathbf{K}$ and $\mathbf{\Sigma}_1$ fulfilling the LMI constraints are found which minimize λ. In practice, the weights in (B.40) should be chosen so that $w_\lambda \gg w_K, w_\Sigma$, because the objective is to fulfill $\lambda = 0$. In the experimental results of this thesis, for example, $w_\lambda = 1 \cdot 10^{10}$, $w_K = 1$, and $w_\Sigma = 1$. The LMIP can be solved efficiently using the tools available for Matlab, such as Yalmip [94] or SeDuMi [91].

Appendix C DBSFM and DDBSFM details

C.1 Inverse optimal control problem

The inverse problem of optimal control arises when the feedback gain matrix or estimator gain matrix is known, but the weights which result in those matrices when solving the optimal control problem are unknown. Given the LTI system

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases}, \tag{C.1}$$

where $(\mathbf{A}, \mathbf{B})$ is stabilizable [24]. The LQR optimal control problem is to solve the control $\mathbf{u}(t)$ which minimizes the performance index

$$J = \int_0^\infty \left(\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}\right) dt, \tag{C.2}$$

where $\mathbf{Q} \geq 0$, $\mathbf{R} > 0$, and $(\mathbf{Q}, \mathbf{A})$ is detectable. The solution of the LQR problem is the state feedback $u(t) = \mathbf{L}\mathbf{x}(t)$, where $\mathbf{L} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$, where $\mathbf{P}$ is the unique positive definite solution of the algebraic Riccati equation

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0 \tag{C.3}$$

The inverse optimal control problem can be posed as: given a gain $\mathbf{L}$, determine $\mathbf{Q} \geq 0$ and $\mathbf{R} > 0$, when $(\mathbf{A}, \mathbf{B})$ is observable and $\mathbf{u}(t) = \mathbf{L}\mathbf{x}(t)$ is the optimal control law of the optimal control minimizing (C.2) [24].

The inverse optimal control problem can be solved using linear matrix inequalities LMIs as follows. Find $\mathbf{R} > 0$, $\mathbf{Q} \geq 0$ and $\mathbf{P} > 0$ so that [92]

$$\begin{aligned} &\min \lambda \text{ s.t.} \\ &\begin{cases} \mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{L}^T\mathbf{R}\mathbf{L} + \mathbf{Q} = 0 \\ \begin{bmatrix} \mathbf{Z} & \mathbf{S}^T \\ \mathbf{S} & \mathbf{I} \end{bmatrix} > 0, \text{where } \mathbf{S} = \mathbf{B}^T\mathbf{P} + \mathbf{R}\mathbf{L},\ \mathbf{Z} < \lambda\mathbf{I} \end{cases}. \end{aligned} \tag{C.4}$$

C.2 Discrete-time inverse optimal control problem

In discrete-time domain, the inverse optimal control problem proceeds similarly as in continuous-time. Given the discrete-time LTI system

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \end{cases}, \tag{C.5}$$

where $(\mathbf{\Phi}, \mathbf{\Gamma})$ is stabilizable. The LQR optimal control problem is to solve the control $\mathbf{u}(k)$ which minimizes the performance index

$$J = \sum_{k=0}^{\infty} \left(\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k) \right), \tag{C.6}$$

where $\mathbf{Q} \geq 0$, $\mathbf{R} > 0$, and $(\mathbf{Q}, \mathbf{\Phi})$ is detectable. The solution of the LQR problem is the state feedback $u(k) = \mathbf{L}\mathbf{x}(k)$, where $\mathbf{L} = -\left(\mathbf{\Gamma}^T\mathbf{P}\mathbf{\Gamma} + \mathbf{R}\right)^{-1}\mathbf{\Gamma}^T\mathbf{P}\mathbf{\Phi}$, where $\mathbf{P}$ is the unique positive definite solution of the algebraic Riccati equation

$$\mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} - \mathbf{L}^T\left(\mathbf{\Gamma}^T\mathbf{P}\mathbf{\Gamma} + \mathbf{R}\right)\mathbf{L} + \mathbf{Q} = \mathbf{P} \tag{C.7}$$

The inverse optimal control problem can be posed as: given a gain $\mathbf{L}$, determine $\mathbf{Q} \geq 0$ and $\mathbf{R} > 0$, when $(\mathbf{\Phi}, \mathbf{\Gamma})$ is observable and $\mathbf{u}(k) = \mathbf{L}\mathbf{x}(k)$ is the optimal control law of the optimal control minimizing (C.6).

Similarly as in the continuous-time case, the inverse optimal control problem can be solved using linear matrix inequalities LMIs as follows. Find $\mathbf{R} > 0$, $\mathbf{Q} \geq 0$ and $\mathbf{P} > 0$ so that

$$\begin{array}{l} \min \lambda \text{ s.t.} \\ \left\{ \begin{array}{l} \mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} - \mathbf{L}^T\left(\mathbf{\Gamma}^T\mathbf{P}\mathbf{\Gamma} + \mathbf{R}\right)\mathbf{L} + \mathbf{Q} = \mathbf{P} \\ \begin{bmatrix} \mathbf{Z} & \mathbf{S}^T \\ \mathbf{S} & \mathbf{I} \end{bmatrix} > 0, \text{where } \mathbf{S} = \left(\mathbf{\Gamma}^T\mathbf{P}\mathbf{\Gamma} + \mathbf{R}\right)\mathbf{L} + \mathbf{\Gamma}^T\mathbf{P}\mathbf{\Phi},\ \mathbf{Z} < \lambda\mathbf{I} \end{array} \right. . \end{array} \tag{C.8}$$

Appendix D Numerical solutions for the structural models

D.1 Linear case MOCM

$$\mathbf{Q}_0 = \begin{bmatrix} 1 & 0 & -0.143404 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.143404 & 0 & 0.303151 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{D.1}$$

$$\mathbf{W}_1 = \begin{bmatrix} 0.143121 & -0.043850 & 0.159846 & -0.000011 & 0 & 0 & 0 \\ -0.043850 & 0.064581 & -0.067129 & 0.000041 & 0 & 0 & 0 \\ 0.159846 & -0.067129 & 0.184980 & -0.000025 & 0 & 0 & 0 \\ -0.000011 & 0.000041 & -0.000025 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.624278 \end{bmatrix} \tag{D.2}$$

$$\tau_n = 0.1125 \tag{D.3}$$

$$f = 5.2094 \cdot 10^{-5} \tag{D.4}$$

$$\mathbf{V}_y = 0.027135 \tag{D.5}$$

$$V_u = 1.624278 \tag{D.6}$$

$$\tau = 0.492350 \tag{D.7}$$

D.2 Linear case HWMOCM

$$\mathbf{Q}_0 = \begin{bmatrix} 1 & 0 & -0.454566 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.454566 & 0 & 0.933768 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{D.8}$$

$$\mathbf{W}_1 = \begin{bmatrix} 3.842694 & 0.029337 & -2.474563 & 0.000009 & 0 & 0 & 0 \\ 0.029337 & 0.094012 & 0.292358 & 0.000002 & 0 & 0 & 0 \\ -2.474563 & 0.292358 & 2.626636 & 0 & 0 & 0 & 0 \\ 0.000009 & 0.000002 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.861485 \end{bmatrix} \tag{D.9}$$

$$\tau_n = 0.100000 \tag{D.10}$$

$$f = 5.2960 \cdot 10^{-5} \tag{D.11}$$

$$\mathbf{V}_y = 0.093961 \tag{D.12}$$

$$V_u = 3.861485 \tag{D.13}$$

$$\tau = 0.400072 \tag{D.14}$$

D.3 Inverse solutions for DBSFM and DDBSFM

The numerical values of the inverse optimal control and estimation solutions for the DBSFM and DDBSFM estimated using the linear HMI data are presented in the following. The numerical values for DBSFM optimal control part are shown in Table D.1 and the optimal estimator part in Table D.2. The numerical values for DDBSFM suboptimal control part are shown in Table D.3 and the suboptimal estimator part in Table D.4.

Table D.1: The solution of the inverse optimal control problem for DBSFM estimated for the linear HMI data.

$$\mathbf{Q}_0 = \begin{bmatrix} 1.000000 & 0.094386 & 0.309596 & 0.021430 & -0.208510 & 0.055704 & 0.002593 \\ 0.094386 & 0.078098 & 0.103587 & 0.044065 & 0.008576 & 0.020209 & -0.000061 \\ 0.309596 & 0.103587 & 0.265221 & 0.048888 & -0.007222 & 0.036152 & 0.000656 \\ 0.021430 & 0.044065 & 0.048888 & 0.027917 & 0.008018 & 0.008823 & 0.000187 \\ -0.208510 & 0.008576 & -0.007222 & 0.008018 & 0.084833 & -0.002604 & -0.002413 \\ 0.055704 & 0.020209 & 0.036152 & 0.008823 & -0.002604 & 0.007540 & -0.000091 \\ 0.002593 & -0.000061 & 0.000656 & 0.000187 & -0.002413 & -0.000091 & 0.000161 \end{bmatrix} \tag{D.15}$$

$$\mathbf{K} = \begin{bmatrix} 0.837142 & 0.354226 & 0.563318 & 0.215698 & -0.088208 & 0.103969 & 0.005853 \\ 0.354226 & 0.221376 & 0.249924 & 0.142526 & -0.041462 & 0.061178 & 0.004641 \\ 0.563318 & 0.249924 & 0.529057 & 0.157006 & -0.035683 & 0.084856 & 0.005335 \\ 0.215698 & 0.142526 & 0.157006 & 0.096545 & -0.020563 & 0.037295 & 0.003423 \\ -0.088208 & -0.041462 & -0.035683 & -0.020563 & 0.022471 & -0.011452 & 0.000100 \\ 0.103969 & 0.061178 & 0.084856 & 0.037295 & -0.011452 & 0.020307 & 0.001343 \\ 0.005853 & 0.004641 & 0.005335 & 0.003423 & 0.000100 & 0.001343 & 0.000310 \end{bmatrix} \tag{D.16}$$

$$\mathbf{N}_0^T = \begin{bmatrix} 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \end{bmatrix} \tag{D.17}$$

$$f = 3.4259 \cdot 10^{-5} \tag{D.18}$$

Table D.2: The solution of the inverse optimal estimation problem for DBSFM estimated for the linear HMI data.

$$\mathbf{W}_1 = \begin{bmatrix} 0.090732 & -1.549549 & -0.049274 & 1.508310 & -0.905463 & 1.124563 & 3.902202 \\ -1.549549 & 26.768478 & 0.846618 & -26.103184 & 15.659961 & -19.291097 & -67.870904 \\ -0.049274 & 0.846618 & 0.033603 & -0.826923 & 0.493187 & -0.627872 & -1.958935 \\ 1.508310 & -26.103184 & -0.826923 & 25.532967 & -15.273414 & 18.705456 & 65.659152 \\ -0.905463 & 15.659961 & 0.493187 & -15.273414 & 9.170758 & -11.269432 & -39.828697 \\ 1.124563 & -19.291097 & -0.627872 & 18.705456 & -11.269432 & 14.143566 & 48.824514 \\ 3.902202 & -67.870904 & -1.958935 & 65.659152 & -39.828697 & 48.824514 & 184.042080 \end{bmatrix} \tag{D.19}$$

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.003202 & -0.024066 & -0.000745 & 0.020940 & -0.017452 & 0.019951 & 0.088034 \\ -0.024066 & 0.719288 & 0.018426 & -0.726944 & 0.418255 & -0.446549 & -1.935113 \\ -0.000745 & 0.018426 & 0.006462 & -0.015649 & 0.007041 & -0.022893 & 0.012879 \\ 0.020940 & -0.726944 & -0.015649 & 0.780315 & -0.413944 & 0.399693 & 1.802966 \\ -0.017452 & 0.418255 & 0.007041 & -0.413944 & 0.267521 & -0.264151 & -1.310761 \\ 0.019951 & -0.446549 & -0.022893 & 0.399693 & -0.264151 & 0.400993 & 1.165761 \\ 0.088034 & -1.935113 & 0.012879 & 1.802966 & -1.310761 & 1.165761 & 8.398043 \end{bmatrix} \tag{D.20}$$

$$\mathbf{N}_1^T = \begin{bmatrix} 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 & 0.000000 \end{bmatrix} \tag{D.21}$$

$$\mathbf{V}_y = 2.4066 \cdot 10^{-4} \tag{D.22}$$

Table D.3: The solution of the inverse optimal control problem for DDBSFM estimated for the linear HMI data.

$$\mathbf{Q}_0 = \begin{bmatrix} 1.000000 & 0.097290 & 0.584521 & 0.084089 & -0.607254 & 0.062908 & 0.022269 \\ 0.097290 & 0.154208 & 0.075828 & 0.094556 & 0.244458 & 0.042768 & -0.003851 \\ 0.584521 & 0.075828 & 0.406834 & 0.064509 & -0.540613 & 0.021277 & 0.024520 \\ 0.084089 & 0.094556 & 0.064509 & 0.061209 & 0.098166 & 0.025551 & -0.000234 \\ -0.607254 & 0.244458 & -0.540613 & 0.098166 & 3.069349 & 0.254544 & -0.139208 \\ 0.062908 & 0.042768 & 0.021277 & 0.025551 & 0.254544 & 0.042108 & -0.013140 \\ 0.022269 & -0.003851 & 0.024520 & -0.000234 & -0.139208 & -0.013140 & 0.007265 \end{bmatrix} \tag{D.23}$$

$$\mathbf{K} = \begin{bmatrix} 11.368135 & 3.656758 & 6.851078 & 2.356876 & -1.395008 & 1.086667 & 0.137997 \\ 3.656758 & 2.380023 & 2.394529 & 1.530859 & -0.267936 & 0.527826 & 0.097027 \\ 6.851078 & 2.394529 & 5.336159 & 1.559798 & -1.021687 & 0.742873 & 0.115370 \\ 2.356876 & 1.530859 & 1.559798 & 1.024949 & -0.216242 & 0.328011 & 0.068007 \\ -1.395008 & -0.267936 & -1.021687 & -0.216242 & 3.727200 & 0.143970 & -0.171407 \\ 1.086667 & 0.527826 & 0.742873 & 0.328011 & 0.143970 & 0.209838 & 0.001123 \\ 0.137997 & 0.097027 & 0.115370 & 0.068007 & -0.171407 & 0.001123 & 0.017921 \end{bmatrix} \tag{D.24}$$

$$\mathbf{N}_0^T = \begin{bmatrix} 0.024933 & 0.017249 & 0.011579 & 0.011766 & 0.080643 & 0.016966 & -0.004394 \end{bmatrix} \tag{D.25}$$

$$f = 0.008666 \tag{D.26}$$

Table D.4: The solution of the inverse optimal estimation problem for DDBSFM estimated for the linear HMI data.

$$\mathbf{W}_1 = \begin{bmatrix} 0.002330 & -0.001883 & 0.003076 & -0.007560 & -0.013359 & 0.001912 & 0.099048 \\ -0.001883 & 0.365754 & 0.033473 & -0.441203 & 0.093493 & -0.300381 & 0.568350 \\ 0.003076 & 0.033473 & 0.010605 & -0.051231 & -0.014606 & -0.040150 & 0.232637 \\ -0.007560 & -0.441203 & -0.051231 & 0.588381 & -0.063583 & 0.332307 & -1.124369 \\ -0.013359 & 0.093493 & -0.014606 & -0.063583 & 0.124083 & -0.043464 & -0.560117 \\ 0.001912 & -0.300381 & -0.040150 & 0.332307 & -0.043464 & 0.343296 & -0.640358 \\ 0.099048 & 0.568350 & 0.232637 & -1.124369 & -0.560117 & -0.640358 & 6.548207 \end{bmatrix} \tag{D.27}$$

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 0.005662 & -0.000779 & -0.000408 & -0.007392 & -0.001838 & 0.028356 & -0.043597 \\ -0.000779 & 0.517818 & 0.053086 & -0.589993 & 0.196096 & -0.352102 & 0.184573 \\ -0.000408 & 0.053086 & 0.032151 & -0.039380 & 0.011870 & -0.092017 & 0.125249 \\ -0.007392 & -0.589993 & -0.039380 & 0.885852 & -0.222412 & 0.172550 & -0.615025 \\ -0.001838 & 0.196096 & 0.011870 & -0.222412 & 0.117387 & -0.131139 & -0.042686 \\ 0.028356 & -0.352102 & -0.092017 & 0.172550 & -0.131139 & 0.774856 & -0.364184 \\ -0.043597 & 0.184573 & 0.125249 & -0.615025 & -0.042686 & -0.364184 & 4.794054 \end{bmatrix} \tag{D.28}$$

$$\mathbf{N}_1^T = \begin{bmatrix} 0.006010 & 0.056356 & 0.015889 & -0.093302 & -0.032064 & -0.058654 & 0.424477 \end{bmatrix} \tag{D.29}$$

$$\mathbf{V}_y = 0.030816 \tag{D.30}$$

Appendix E Optimal cost function simulation figures

In this appendix, the simulation results for the optimal tuning experiments are shown. The result given by the optimal tuning is studied by varying the noise parameters of the MOCM described in Appendix D.1. The parameters are varied by multiplying the matrices with a scalar ranging from 0.1 to 10. This corresponds to changing the relative noise level of the plant, measurement process, and the neuromotor. In the figures below, the responses given by the MOCMs with varied parameters are plotted as blue. For a single set of noise parameters, several relative gain values from 0.2 to 1.2 are used. The response with the optimal relative gain k_n/k_0 value given by the optimal tuning method is plotted as red.

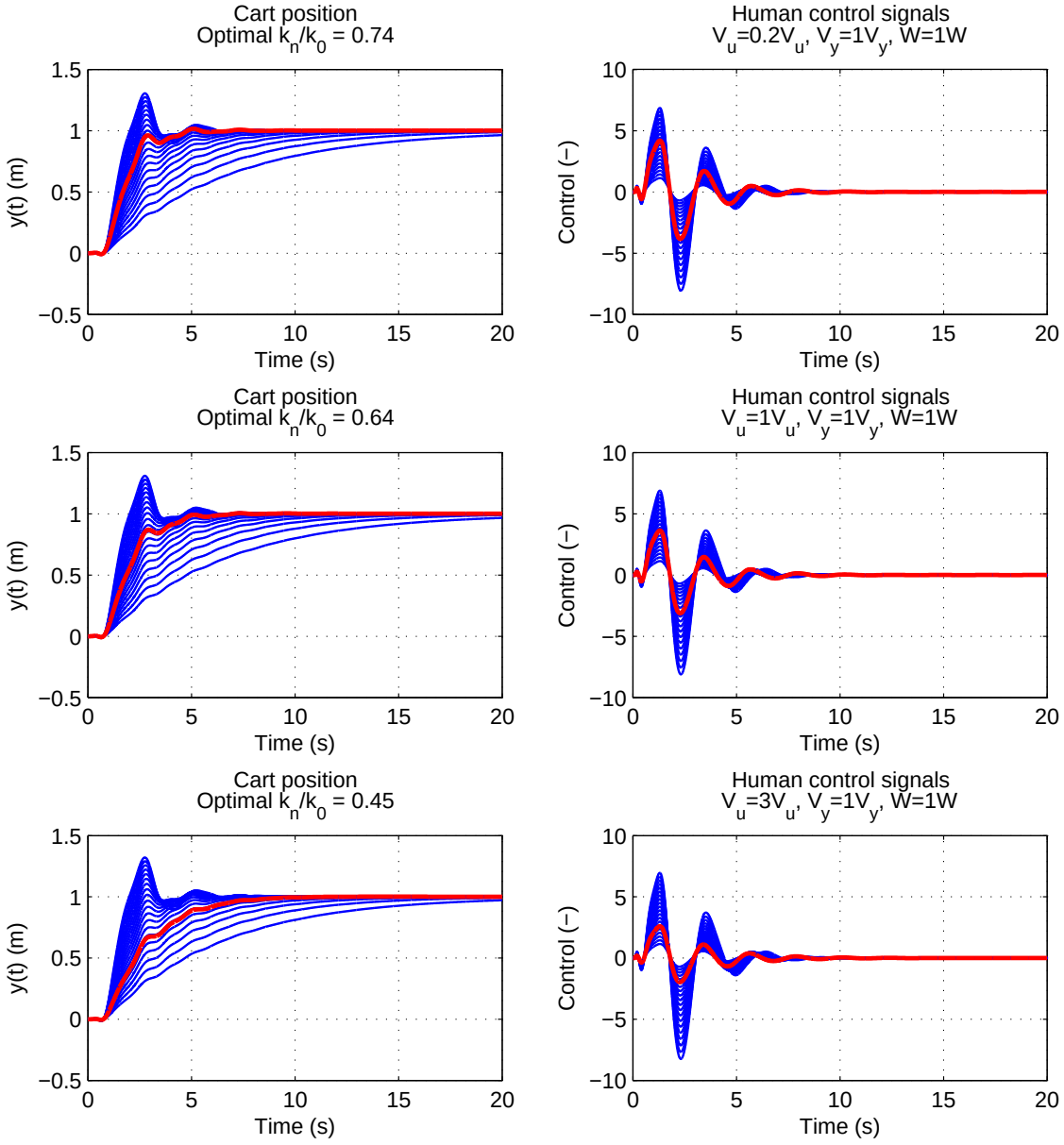


Figure E.1: Optimal cost function simulation when V_u is varied for MOCM. The response given by the optimal relative gain value is shown red.

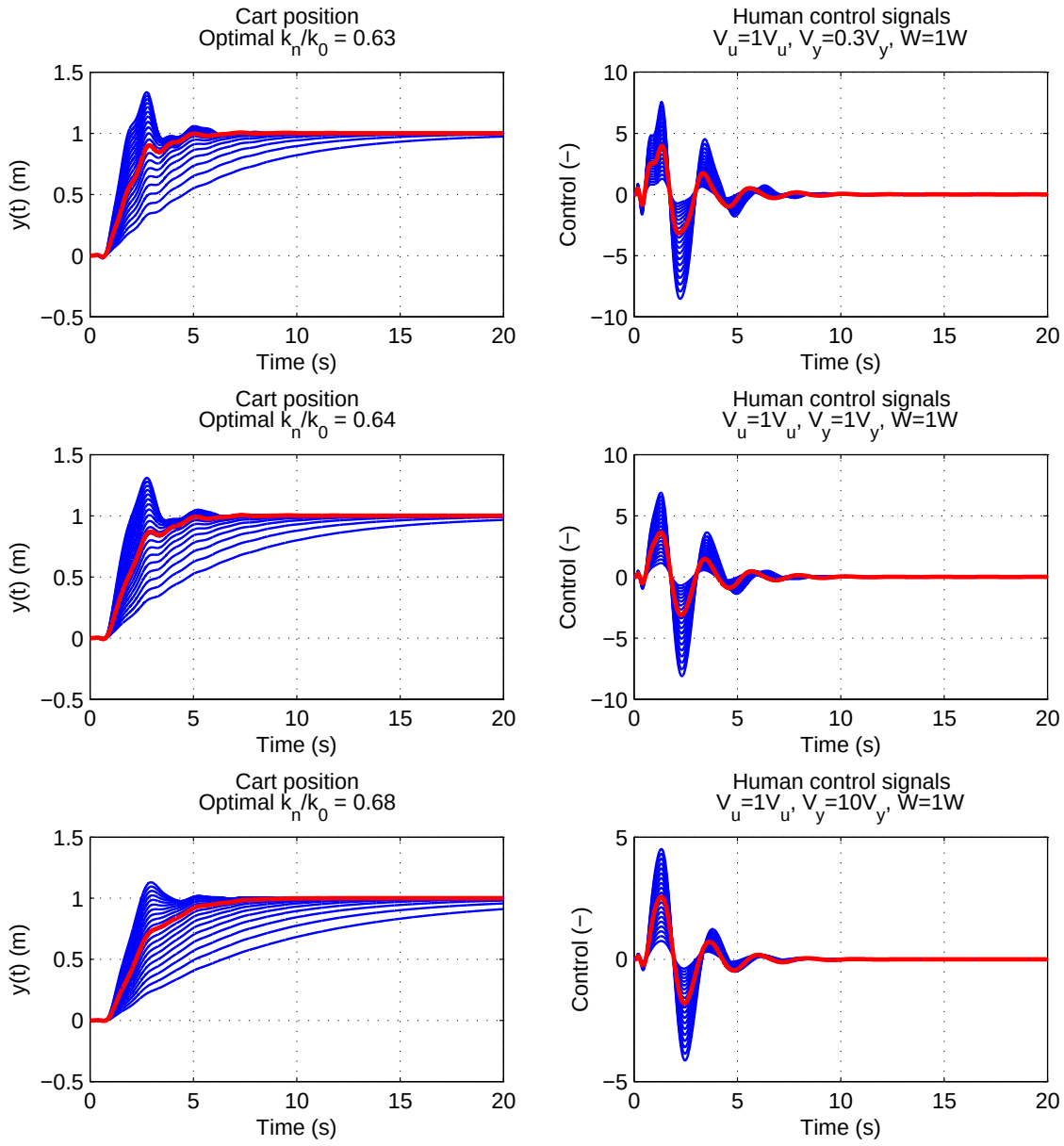


Figure E.2: Optimal cost function simulation when $\mathbf{V}_y$ is varied for MOCM. The response given by the optimal relative gain value is shown red.

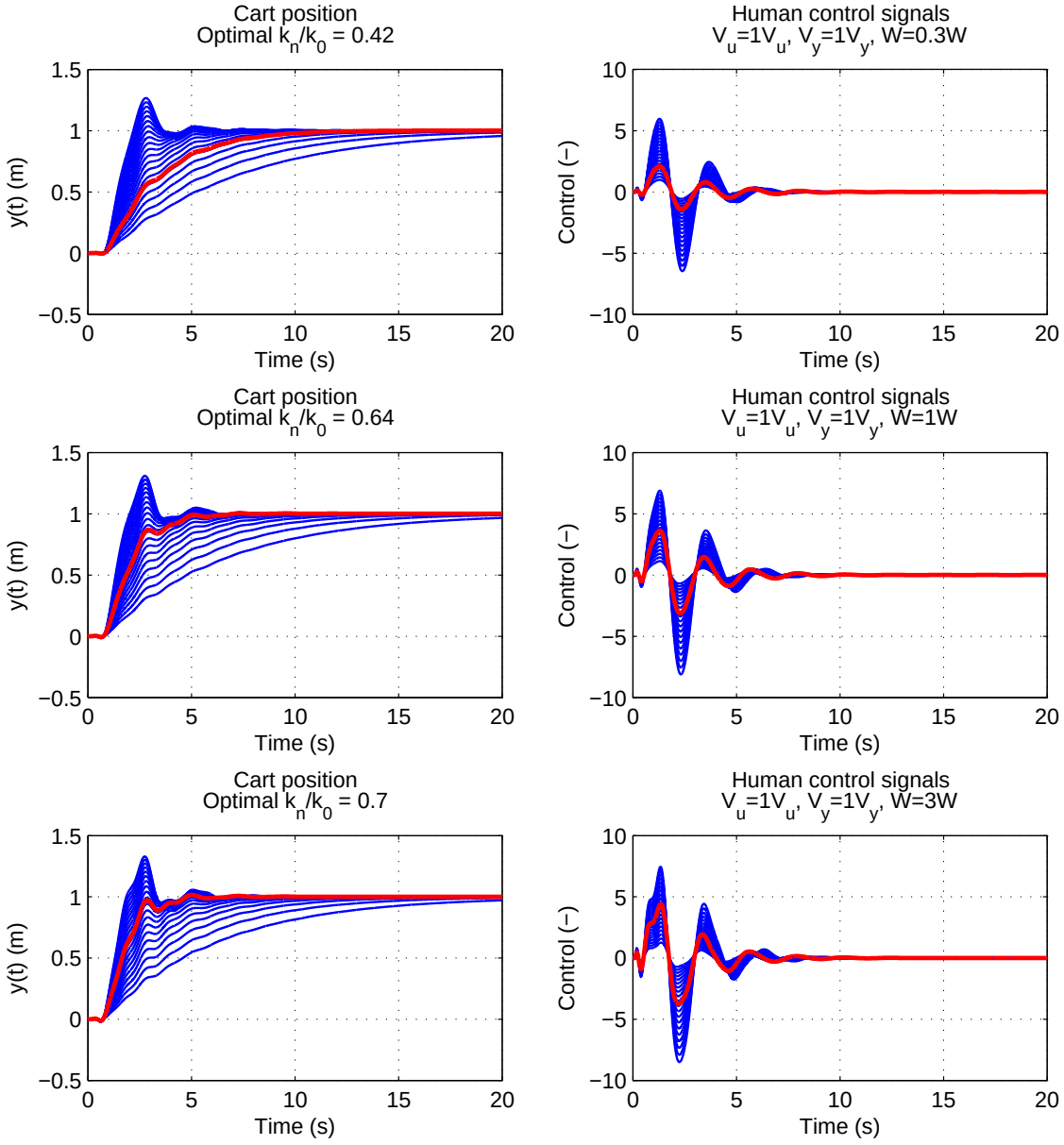


Figure E.3: Optimal cost function simulation when **W** is varied for MOCM. The response given by the optimal relative gain value is shown red.

HELSINKI UNIVERSITY OF TECHNOLOGY CONTROL ENGINEERING

Editor: H. Koivo

Report 156 Korkiakoski, V.
Improving the Performance of Adaptive Optics Systems with Optimized Control Methods. April 2008.

Report 157 Al.Towati, A.
Dynamic Analysis and QFT-Based Robust Control Design of Switched-Mode Power Converters. September 2008.

Report 158 Eriksson, L.
PID Controller Design and Tuning in Networked Control Systems. October 2008.

Report 159 Pohjoranta, A.
Modelling Surfactant Mass Balance with the ALE Method on Deforming 2D Surfaces. May 2009.

Report 160 Kaartinen, J.
Machine Vision in Measurement and Control of Mineral Concentration Process. June 2009.

Report 161 Hölttä, V.
Plant Performance Evaluation in Complex Industrial Applications. September 2009.

Report 162 Halmevaara, K.
Simulation Assisted Performance Optimization of Large-Scale Multiparameter Technical Systems. September 2009.

Report 163 Haavisto, O.
Reflectance Spectrum Analysis of Mineral Flotation Froths and Slurries. November 2009.

Report 164 Cosar, E. I.
A Wireless Toolkit for Monitoring Applications. October 2009.

Report 165 Pohjoranta, A.
Computational Models for Microvia Fill Process Control. February 2010.

Report 166 Mendelson, A.
Identification and Control of Deposition Processes. March 2010.

Report 167 Björkbom, M.
Wireless Control System Simulation and Network Adaptive Control. October 2010.

Report 168 Tervo, K.
Human Adaptive Mechatronics Methods for Mobile Working Machines. November 2010.

ISBN 978-952-60-3529-1 (printed)

ISBN 978-952-60-3530-7 (pdf)

ISSN 0356-0872

Aalto-Print, Helsinki 2010